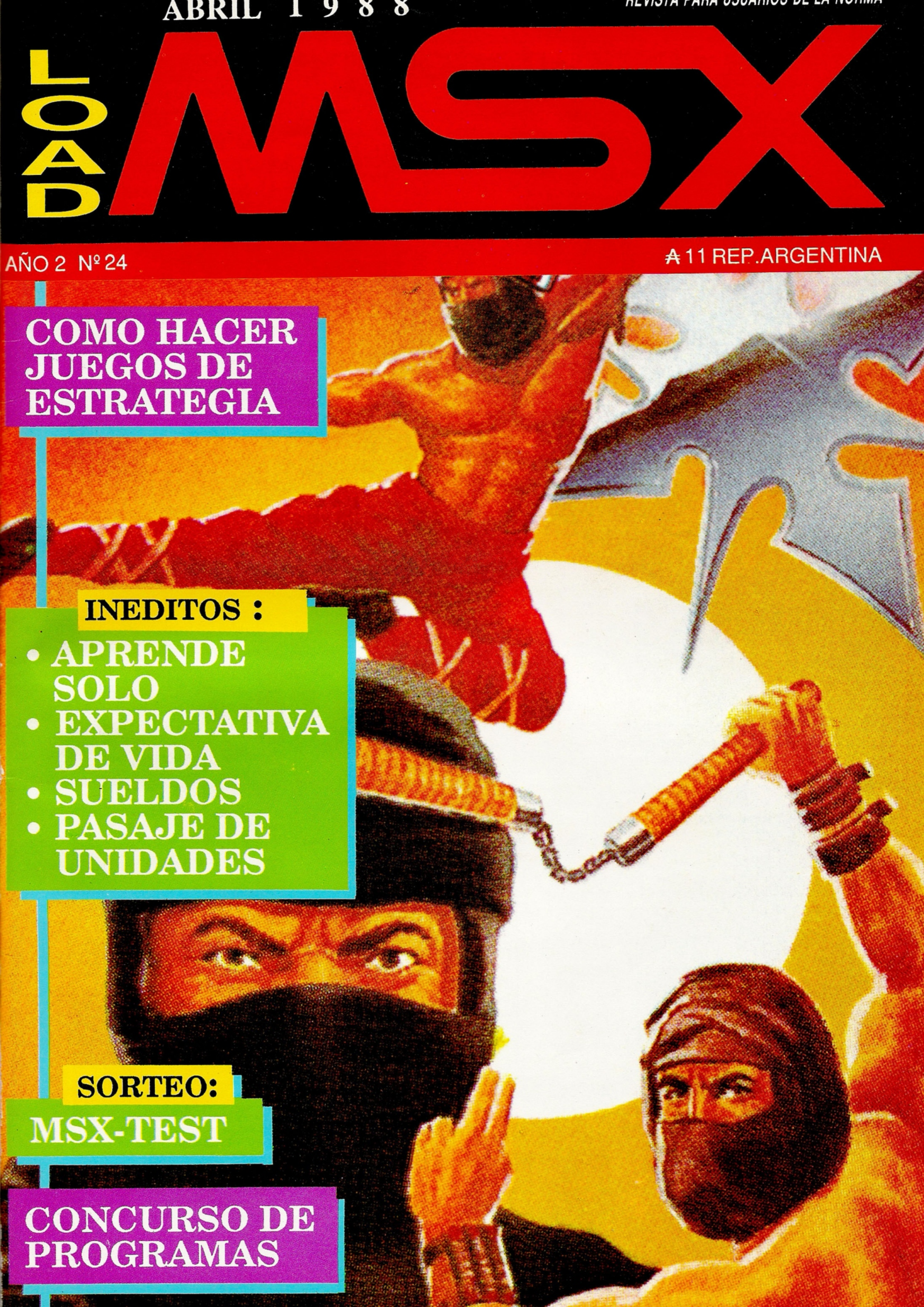


ABRIL 1988
REVISTA PARA USUARIOS DE LA NORMA
LOAD
MSX
AÑO 2 Nº 24
₳ 11 REP.ARGENTINA
COMO HACER JUEGOS DE ESTRATEGIA
INEDITOS :
• APRENDE SOLO
• EXPECTATIVA DE VIDA
• SUELDOS
• PASAJE DE UNIDADES
SORTEO:
MSX-TEST
CONCURSO DE PROGRAMAS

STAFF

LOAD MSX

Director Periodístico

Fernando Flores

Secretario de Redacción

Ariel Testori

Redacción

Fernando Pedró

Arte y Diagramación

Fernando Amengual y Tamara Migelson

Departamento de Avisos

Oscar Devoto y Nelzo Capello

Departamento de Publicidad

Dolores Urien

Servicios Fotográficos

Image Bank, Oscar Burriel y Victor Grubicy

EDITORIAL PROEDI

Presidente

Ernesto del Castillo

Vicepresidente

Cristian Pusso

Director Titular

Javier Campos Malbrán

Director Suplente

Armengol Torres Sabaté

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

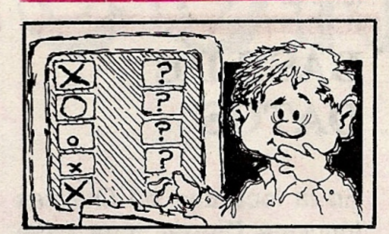
ISSN 0326-8241
Impreso en Impresiones Gráficas Tabaré S.A.I.C. Erézcano 3158 Cap. - Fotocromo tapa: Columbia.

Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital.
Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

CORREO ARGENTINO CASA CENTRAL | FRANQUEO PAGO CONCESION Nº [illegible] | TARIFA REDUCIDA CONCESION Nº 1304

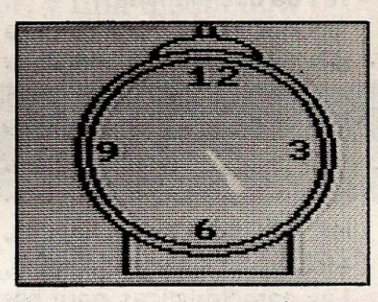
SUMARIO

JUEGOS DE ESTRATEGIA



Ante una jugada nuestra, la computadora responde con otra que puede ponernos en un aprieto o directamente vencernos. ¿Pero cómo hace la máquina para evaluar su táctica a seguir? (Pág. 6)

RELOJ DIGITAL

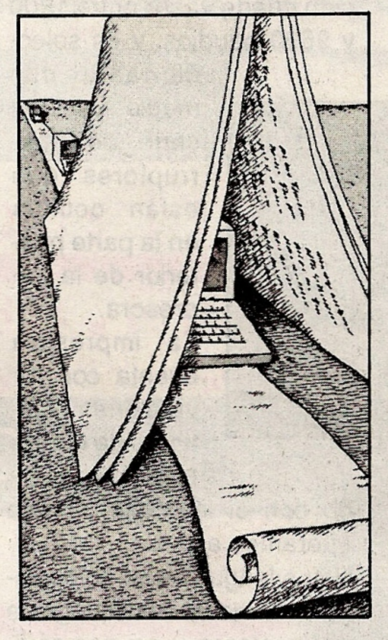


Con este recurso podremos utilizar la computadora con precisión más aceptable, tanto para medir la duración de sucesos externos al sistema, como para cronometrar determinados algoritmos que se encargan de tareas comunmente lerdas, tales como el ordenamiento de vectores y búsquedas. (Pág. 10)

RINCON DEL USUARIO

Los interrogantes sobre el manejo de archivos desde el MSX Disk BASIC se podrían resumir en uno solo: el manual no especifica con un ejemplo que englobe todo lo referente al tema y, si se quiere, no trae un curso sobre el mismo. (Pág. 20)

LA AVENTURA DEL PENSAMIENTO

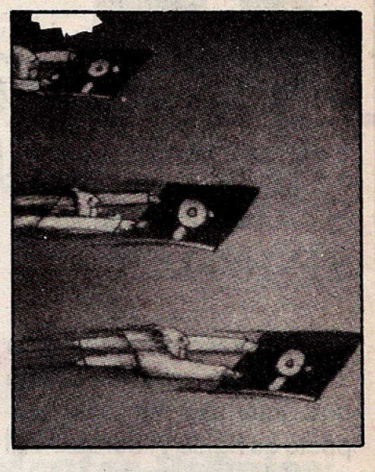


El desarrollo de software requiere de un proceso de varias etapas. A través de un ejemplo, un programa que hace gráficos de barras, ofrecemos las experiencias de un grupo de jóvenes sobre el tema. (Pág. 26)

LOS MISTERIOS DEL MSX-DOS

(3ra. parte)

Seguimos investigando el MSX-DOS para poner luz sobre esta importante herra-



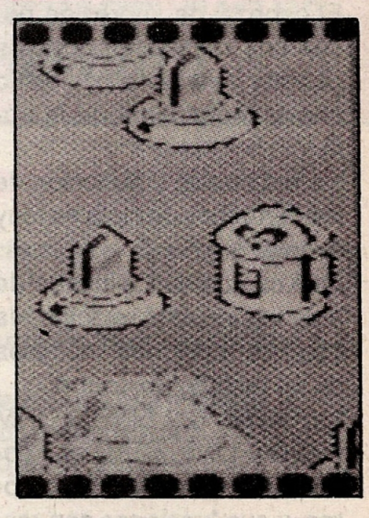
mienta. Analizaremos el tema del control de archivos. (Pág. 28)

PROGRAMAS



Sueldos (Pág. 12) - Cálculo de expectativa de vida (Pág. 16) - Pasaje de unidades (Pág. 19) - Aprende solo (Pág. 25)

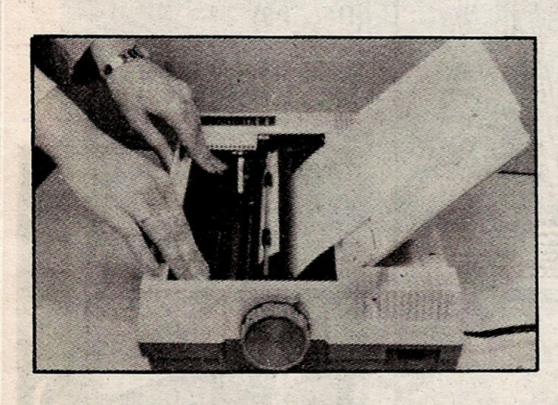
SECCIONES FIJAS



Noticias MSX (Pág. 4) - Sortilegios (Pág. 31) - Soft al día (Pág. 32) - Buzón (Pág. 34)

IMPRESORA SEIKOSHA SP-1000AS

Esta impresora funciona en todas las computadoras con una interfase serie.Su aspecto es rígido y "durable". El panel de control que se ubica a la izquierda de la misma incluye todos los comandos habituales, más uno extra, que nos permite poner a la impresora a trabajar en modo NLQ (Near Letter Quality), Este nos brinda una calidad de impresión comparable a la de una máquina de escribir.



Otra característica interesante es el sistema de transporte del papel.
El mismo incluye una bandeja que se eleva para separar convenientemente el papel que entra a la impresora de aquel que sale ya impreso.
El cabezal de impresión posee una regulación que nos permite acercarlo o alejarlo convenientemente de la hoja.
La instalación y remoción de los cartuchos de cinta es muy simple.
Otra característica inusual de este equipo está relacionada con las funciones múltiples de su teclado de control.
Las teclas ON LINE, FORM FEED y LINE FEED son comunes a todas las impresoras, y sus funciones son respectivamente poner a la impresora en servicio, avanzar una hoja completa de papel y avanzar una línea.
Sin embargo, estas tres teclas cumplen otra función importante que es la de establecer los márgenes de impresión.
En cuanto a la interfase que equipa a este modelo, se trata de una versión serie que cumple con el protocolo RS 232.
La velocidad de comunicación puede variar entre 1200 y 9600 baudios, y es seleccionable por medio de una serie de interruptores que están ocultos en la parte posterior de la impresora.
La impresora cuenta con 11 versiones distintas de caracteres de acuerdo con el lenguaje en que queramos escribir.
Estos lenguajes se seleccionan por medio de un juego de interruptores y no por códigos de control.
En cuanto a la velocidad de impresión, tiene un máximo de 100 caracteres por segundo.
Sin embargo, esta velocidad disminuye notablemente en otros modos.
La cantidad máxima de caracteres por línea es de 137, trabajando en modo condensado.
Además del modo de carácter estándar, podemos optar por imprimir con el juego de caracteres itálicos cursivos.
También podemos regular por medio de soft el espaciado entre renglones, y trabajar en modo subrayado y con caracteres en negrita.
Las posibilidades de impresión incluyen un modo gráfico de simple y doble resolución. En este modo, podemos graficar con una resolución de 960 puntos por columna.
Finalmente, un modo especial de cuádruple densidad de impresión nos permite trabajar con 1920 puntos por columna.

BASE DE DATOS

En la localidad de Quilmes funciona **Digger**, una base de datos de características novedosas y completas para la comunidad.
La computadora que se usa en **Digger** es una IBM XT de 640 Kb. y un disco duro de 30 Mb. El programa, que fue desarrollado en su totalidad por los técnicos de la base de datos, permite acceder a una gran gama de opciones a través de sus menúes.
Algunos de los servicios que brinda son: mailing, correo electrónico, colocar avisos clasificados, colocar publicidad, juegos, servicios para la comunidad (hospitales, hoteles, registros civiles, bancos, etcétera), catálogo de video.
Las características para comunicarse son las siguientes: 300 baudios, 8 bits de palabra, 1 bit de stop, sin paridad y normal Bell. El teléfono para entrar en Digger es 254-9008.

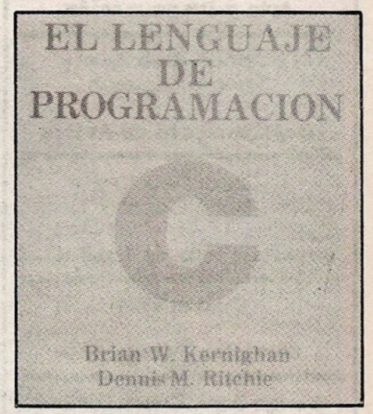
PRIMER GANADOR



El **MSX TEST** demostró haber tenido una gran recepción entre los lectores.
Prueba de ello es la cantidad de respuestas recibidas para los diferentes Test en curso.
El **MSX TEST** nº 1, que cerró el 1º de marzo, ya tiene su ganador. Es **Rubén Papazian**, tiene 35 años, vive en la ciudad de Cosquín, Córdoba, y es poseedor de una Talent MSX.
Con su ingenio ganó un soft a elección entre el Idea Base, Idea Text o Basic Tutor.

EL LENGUAJE DE PROGRAMACION C

AUTOR: Brian W. Kernighan- Dennis M.Ritchie



Este libro se propone ayudar al lector a aprender a programar en C.
Este lenguaje felizmente se encuentra disponible para computadoras hogareñas como la MSX y la mayoría de las computadoras de la línea PC.
El libro no es un manual introductorio a la programación, sino que supone conocimientos de programación en algún lenguaje.
Esto no significa que el programador principiante encuentre dificultades al leer y comprender sobre la marcha.
El primer capítulo es una introducción general a la parte central de C.
Los capítulos 2 a 6 desarrollan algunos aspectos del C con más detalle como: "operadores, expresiones, estructura del programa, variables externas, apuntadores, aritmética de direcciones, estructuras y uniones".
El capítulo 7 describe la biblioteca básica de entrada y salida de datos en C.
Por último, el octavo capítulo describe las relaciones entre los programas en C y el sistema operativo UNIX. (Distribuye: CUSPIDE).

4to CONCURSO
DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, y una disquetera y una impresora).

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una disquetera y una impresora).

ESPECIAL

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

BASES

No sólo será indispensable que el programa enviado en casete ó disquete funcione correctamente, sino que además debe cumplir con ciertas reglas:
.Programación estructurada en bloques fácilmente diferenciables.
.Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de linea que los identifiquen).
.Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
.Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
.Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
.Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de julio próximo (cierre del certámen) a: Paraná 720, piso 5, (1017) Capital Federal.

JUEGOS DE ESTRATEGIA

Ante una jugada nuestra, la computadora responde con otra que puede ponernos en un aprieto o directamente vencernos. ¿Pero cómo hace la máquina para evaluar su táctica a seguir?

Desde el simple Ta-Te-Ti hasta el muy complicado ajedrez, los juegos de salón son verdaderas "batallas" en las que dos o más jugadores compiten buscando demostrar la superioridad de sus estrategias. Por sus características la computadora es capaz de reemplazar a uno de los competidores y, de acuerdo con la capacidad del programador, tendrá un determinado nivel de juego. Es importante destacar que si bien en los juegos que tienen estrategias difíciles de definir, como el ajedrez, el ordenador esta aún por debajo del nivel humano, en aquellos otros en donde hay una estrategia definida y hace falta una gran memoria y velocidad de cálculo podemos decir que "el programa supera a su programador".



Se puede hacer una clasificación de los juegos de estrategia de acuerdo con dos factores. La información que se posee puede ser perfecta (si no hay movidas simultáneas y ambos jugadores conocen en todo momento mientras dura el juego toda movida anterior) o no (cuando no se cumple alguna de las condiciones). La otra distinción posible es la de los juegos que tienen movimientos de azar, es decir, movimientos que no hacen ninguno de los jugadores, sino algún mecanismo de azar externo, por ejemplo el reparto de cartas.

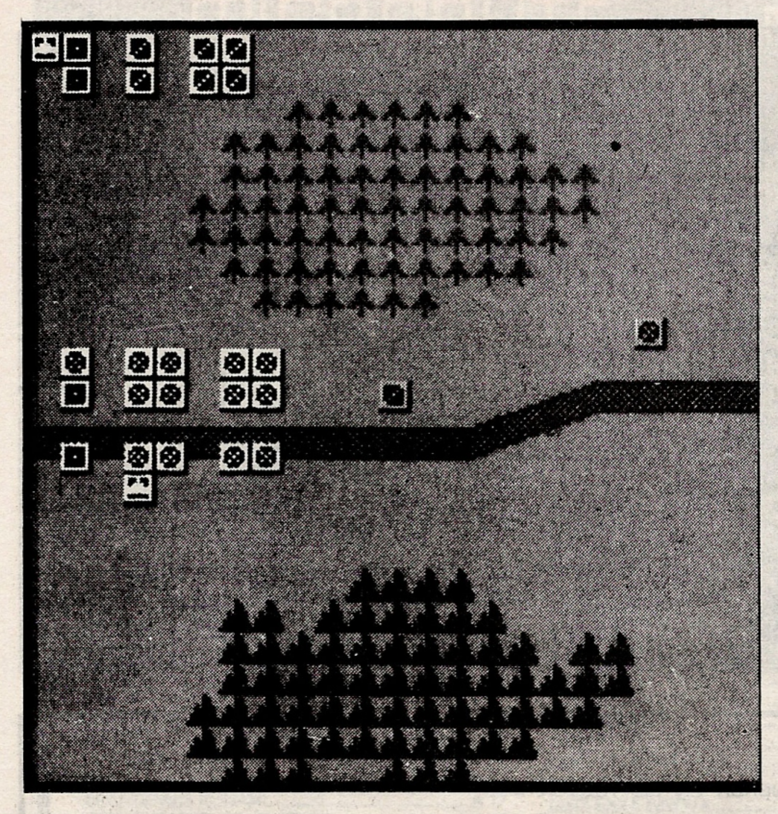
Con estos factores son posibles las combinaciones que ejemplificamos en la figura 1.

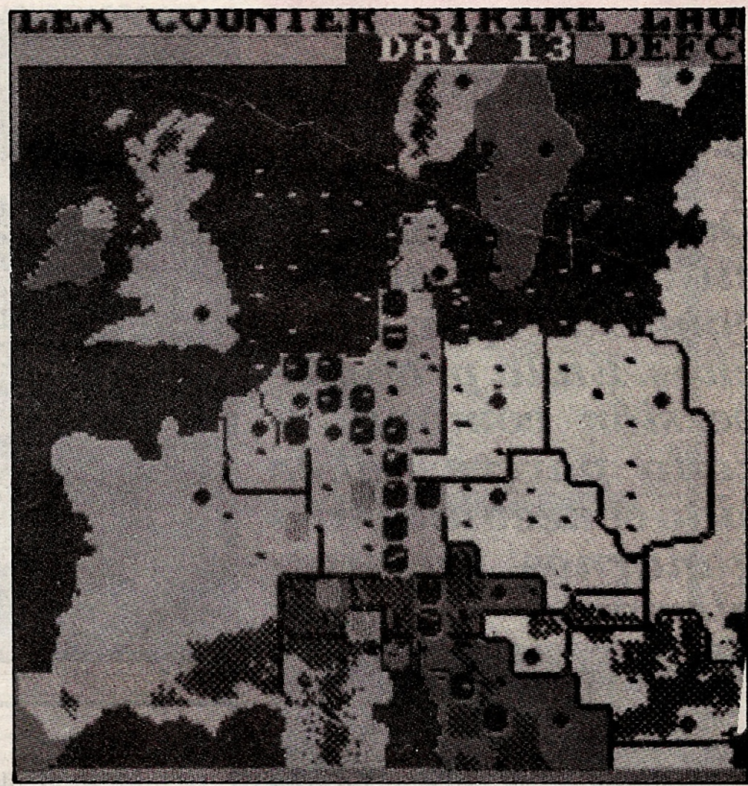
Antes de comenzar un programa de algún juego de estrategia hay que determinar a qué tipo pertenece. Esto nos dará una idea clara de qué factores de evaluación tenemos que ponderar. Analizando el juego en profundidad iremos descubriendo las diferentes estrategias del mismo. Un consejo útil es ir anotando las distintas ideas de estrategias aunque pensemos que sean malas, ya que después se pueden combinar varias estrategias y la que sola era mala, resulta ser muy buena en grupo. ¿Cómo se logra esto? La respuesta a esta pregunta no es una sola, pero mencionaremos aquí a modo de ejemplo la existencia de un polinomio de evaluación, en el que el resultado de cada estrategia es multiplicado por un coeficiente, a criterio del programador, para formar así el término de esa estrategia en partícular. Sumando todos los terminos lograremos una evaluación general en que se han tenido en cuenta varias ideas que, sueltas, no harían nada. También se puede hacerlas actuar en forma independiente eligiendo luego una de las opciones. Esto nos permitirá a la hora de programar disponer de más elementos. Vamos ahora a desarrollar estrategias generales para distintos tipos de juegos y, en notas posteriores, iremos dando trucos y rutinas que permitirán la aplicación de las mismas. En los juegos con información perfecta y con movimientos del azar (los juegos de cartas en los que no hay envites, los de dados) suelen ser muy útil usar la memoria de la computadora para almacenar los datos disponibles y usar su "fuerza bruta" en los cálculos de probabilidades para las diferentes manos. En estos juegos interviene muy poco la estrategia a adoptar por el segundo jugador. En cambio en los juegos de movimiento simultáneo se debe tener en cuenta esa estrategia. Por ejemplo en el papel-piedra-tijeras, el jugador humano, por lo general, se inclina por uno de los tres objetos en desmedro de los otros dos. El ordenador puede aprovechar esto haciendo que su opción más frecuentemente empleada sea la que derrote a la preferida del rival.

La dificultad de los juegos de información perfecta sin movimientos de azar es descubrir la estrategia correcta. Si lo logramos, podremos asegurar que la computadora jugará bien. En estos juegos de tablero se hace una búsqueda de las distintas piezas jugadas y se efectúa una evaluación relacionándolas entre sí. El último factor a considerar para desarrollar un programa de juegos es el elemento físico. Esto no solo tiene que ver con la presentación del juego en sí. En los jue-

CUADRO 1

	INFORMACION PERFECTA	INFORMACION IMPERFECTA
SIN MOVIMIENTO DE AZAR	AJEDREZ	PAPEL-PIEDRA-TIJERAS
CON MOVIMIENTO DE AZAR	LUDO	POKER





gos de cartas, por ejemplo, debemos observar cómo mezclamos el mazo y cómo extraemos las cartas, dónde se van a almacenar los datos de los diferentes pozos, cartas en la mano de la computadora, cartas en la mesa. De esta manera tenemos expuestos someramente los diferentes tipos de juegos de estrategia que existen, lo que nos brinda la base para ir desarrollando este tema con profundidad.

A continuación presentamos un programa que es una muestra del nivel de razonamiento estratégico aplicado a una computadora.

Fernando Pedró

CONCURSO

EL PROGRAMADOR DEL AÑO



BASES

Una vez terminado y revisado tu programa, deberás enviarlo a la editorial grabado en un cassette o diskette, varias veces para mayor seguridad. (Inclusive grabado con dos grabadores distintos). Indicar en el cassette o diskette, los datos del programa, computadora y autor.
Otra condición es que sea original e inédito, es decir que no haya sido enviado a ninguna otra publicación.
Si bien es preferible que vaya acompañado del listado del mismo por impresora, éste no es imprescindible.
El programa deberá venir con un texto que aclare cuál es su nombre, objetivo, modo de uso, y explicación de cada una de sus partes, subrutinas y variables. Si posee lenguaje de máquina, es fundamental una buena explicación sobre su funcionamiento e ingreso a la máquina. No olvidarse los datos completos del autor o autores.
El texto se presentará en hojas tipo oficio y mecanografiado a doble espacio. No importa que la redacción no sea muy clara, eso queda por nuestra cuenta.

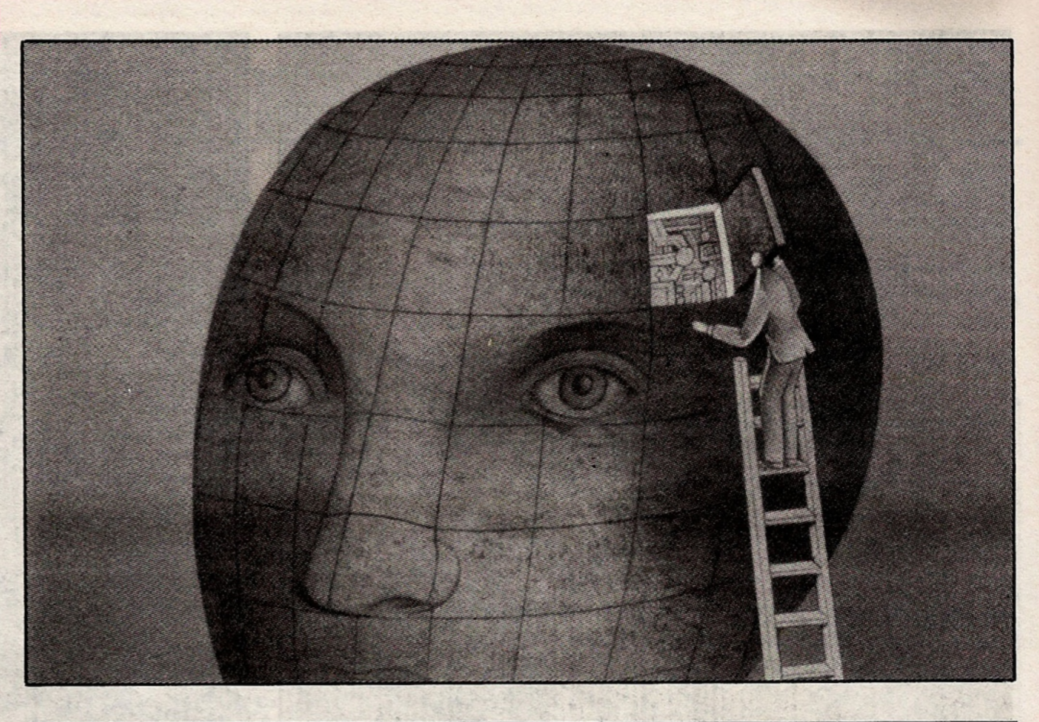
IMPORTANTES PREMIOS

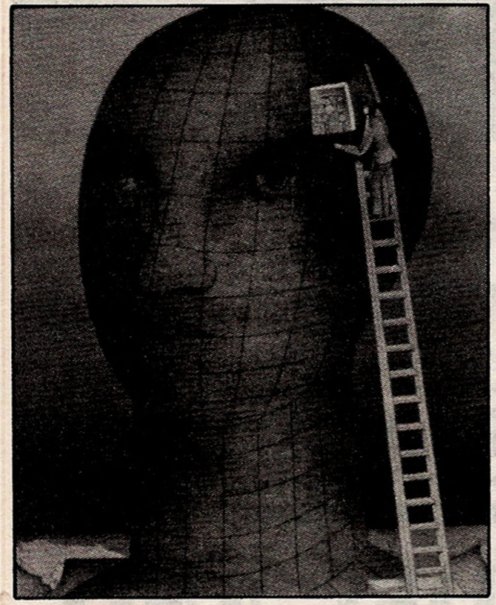
Los daremos a conocer próximamente.

CIERRE: El cierre de recepción de trabajos para concurso de programas será el 30/9/88 . (K64 se reserva el derecho de publicación de los programas recibidos, como asimismo la devolución del material).

NIM

Describimos un sencillo juego, quizás conocido por muchos. En la presente versión computarizada parece haber sido concebido para ganarnos incansablemente.





En efecto, la utilización de configuraciones leídas de instrucciones DATA establecidas de antemano le permiten a la computadora efectuar siempre la jugada precisa para llegar triunfante al final de las partidas. Esta es una muestra del nivel de razonamiento estratégico, en este caso muy elemental, inyectable en nuestra minicomputadora.

Al comenzar el juego se cargan dichas jugadas en un vector que se utiliza sucesivamente a lo largo del programa para tantear por búsqueda secuencial una respuesta ganadora al movimiento humano. Las subrutinas utilizadas en el programa se han ubicado al comienzo del mismo para lograr una mayor rapidez en la búsqueda de la respuesta, y como muestra de que no necesariamente deben agruparse sobre el fin de los programas.

El juego en sí consiste en retirar un número de palillos, por cada turno de juego, variable desde uno hasta todos los de una fila. El participante que logre dejar uno solo sobre la mesa habrá ganado.

```
5 KEY OFF
10 GOTO 210
20 ' COPIA DE JUGADA EN EL VECTOR W
30 FOR K=1 TO 5
40     W(K)=S(K)
50 NEXT K
60 RETURN
70 ' ORDENAMIENTO DEL VECTOR W
80 FOR K=1 TO 5
90     FOR J=K+1 TO 5
100                 IF W(J)>W(K) THEN C=W(J):W(J)=W(K):W(K)=C
110      NEXT J
120 NEXT K
130 RETURN
140 ' TRANSFORMA MATRIZ EN STR$
150 C$=""
160 FOR K=1 TO 5
170      IF W(K)=0 THEN K=5:GOTO 190
180      C$=C$+STR$(W(K))
190 NEXT K
200 RETURN
210 DATA " 1"," 1 1 1"," 1 1 1 1 1"," 2 2"," 2 2 1 1"," 2 2 2 2"
220 DATA " 3 2 1"," 3 2 1 1 1"," 3 2 2 2 1"," 3 3"," 3 3 1 1"
230 DATA " 3 3 2 2"," 3 3 3 2 1"," 3 3 3 3"," 4 4"," 4 4 1 1"
240 DATA " 4 4 2 2"," 4 4 3 2 1"," 4 4 3 3"," 4 4 4 4"," 5 4 1"
250 DATA " 5 4 1 1 1"," 5 4 2 2 1"," 5 4 3 2"," 5 4 3 2 1"
260 DATA " 5 4 4 4 1"," 5 5"," 5 5 1 1"," 5 5 2 2"
270 DATA 5,5,3,2,1
280 DIM D$(29),S(6),W(6):FALSO=0
290 RESTORE 210
300 FOR I=1 TO 29:READ D$(I):NEXT I
310 FOR I=1 TO 5:READ S(I):NEXT I
320 PRINT "DISPOSICION ACTUAL DEL TABLERO :"
330 FOR I=1 TO 5:PRINT "FILA ";I;" CON ";S(I)," PALILLOS":NEXT I
340 INPUT "DE CUAL FILA REMUEVES ";L
350 IF L<1 OR L>5 THEN 340
360 INPUT "CUANTOS PALILLOS          ";NC
370 IF NC<1 OR NC>S(L) THEN 360
380 S(L)=S(L)-NC
390 GOSUB 20
400 FOR I=1 TO 5
410     IF S(I)=0 THEN 540
420     FOR P=1 TO S(I)
430             W(I)=W(I)-P
440             GOSUB 70
450             GOSUB 140
460             ENCUENTRO=FALSO
470             FOR K=1 TO 29
480                 IF C$=D$(K) THEN ENCUENTRO=NOT FALSO
490             NEXT K
500             IF ENCUENTRO THEN NC=P:L=I:P=S(I):GOTO 520
510             GOSUB 20
520     NEXT P
530     IF ENCUENTRO THEN 550
540 NEXT I
550 IF NOT ENCUENTRO THEN PRINT "NO ENCUENTRO SOLUCION POSIBLE":STOP
560 PRINT "MI JUGADA CONSISTE EN SACAR ";NC;" PALILLOS DE LA FILA ";L
570 S(L)=S(L)-NC
580 IF C$<>" 1" THEN 320
590 PRINT "DADO QUE SOLO RESTA UN PALILLO EN LA MESA , HE GANADO!"
```

MICROBYTE Software

MONTEVIDEO 252 - BUENOS AIRES - T.E. 38-0331

SERVICIO PUERTA A PUERTA MSX

ACCESORIOS

JOYSTICKS

ARCADE c/autodisparo	A 115
200X (microswitch)	A 175
V6.747 (palanca al piso)	A 120
V6.123	A 92
Cable grabador MSX	A 25
Diskette 5 1/4 2D.2D x 10	A 75

HARDWARE

Consúltenos por cualquier equipo o periférico

0801-ATHLETIC LAND	A 17
0802-HYPER SPORT 1	A 17
0803-HYPER SPORT 2	A 17
0806-ROAD FIGHTER	A 17
0809-FLIGHT PATH 737	A 17
0814- YIE AR KUNG FU-	A 17
0826-KONAMI'S SOCCER	A 17
0827-BOULDER DASH	A 17
0836-STOP THE EXPRESS	A 17
0837-RIVER RAID	A 17
0841-H.E.R.O.	A 17
0847-TENNIS KONAMI	A 17
0850-ANTARTIC ADVENTURE	A 17
0856-F-16	A 17
0874-PINGUILANDIA	A 17
0875-CHOPLIFTER	A 17
0876-GALAGA	A 17
0877-KUNG FU MASTER	A 17
0892-GHOSTBUSTERS	A 17
0898-PAC-MAN	A 17
0904-EGGERLAND MISTERY	A 17
0895-EXERION I	A 17
0908-PIPPOLS	A 17
0920-THE DAM BUSTERS	A 17
0922-GRAND PRIX	A 17
0924-PING PONG	A 17
0928-THUNDER BALL	A 17
0929-COMIC BAKERY	A 17
0931-BOUNDER	A 17
0932-KNIGHT LORE	A 17
0933-NIGHT SHADE	A 17
0935-TIME PILOT	A 17
0937-VALKYR	A 17
0938-ALIEN 8	A 17
0939-GUN FRIGHT	A 17
0941-KING'S VALLEY	A 17
0942-MAGICAL TREE	A 17
0945-FORMATION Z	A 17
0947-MOON PATROL	A 17
0948-ZAXXON II	A 17
0949-ELEVATOR ACTION	A 17
0950-BASEBALL	A 17
0951-YIE AR KUNG FU II	A 17
0953-LAS TRES LUCES DE GLAURUNG	A17
0954-T.Z.R.	A 17
0955-SCARLET 7	A 17
0960-ZANAC	A 17
0961-EXOIDE Z	A 17
0962-RAID ON BUNGELIN BAY	A 17
0963-SWEET ACORN	A 17
0964-XIXOLOG	A 17
0965-CIRCUS CHARLIE	A 17
0966-HYPER RALLY	A 17
0967-BOSCONIAN	A 17
0968-KNIGHTMARE	A 17
0969-Mr. DO	A 17
0970-PINKY CHASE	A 17
0971-SCION	A 17
0973-SAMURAI NINJA II	A 17
0975-BRUCE LEE	A 17
0976-GYRODINE	A 17
0977-THE WAY OF THE TIGERS	A 48
0978-THE GOONIES	A 17
0979-SKYGALDO	A 17
0980-LAZY JONES	A 17
0981-BLACK JACK	A 17
0983-DUNKSHOT	A 17
0984-B.C.QUEST II	A 17
0985-COASTER RACE	A 17
0986-THEXDER	A 17
0987-EXXA INNOVA	A 17
0989-BANK PANIC	A 17
0990-ZEXXAS II	A 17
0991-GUARDIC	A 17
0992-HANG ON	A 17
0994-ALCAZAR	A 17
0995-LODE RUNNER II	A 17
0997-HOLE IN ONE PROFESSIONAL	A17
0998-RABBIAN	A 17
0999-RAMBO	A 17
1000-STAR FORCE	A 20
1001-MAGICAL KID WIZZARD	A 17
1003-ARKANOID	A 20
1004-GREEN BERET	A 20
1006-MOLE MOLE II	A 17
1007-BATMAN	A 20
1008-HEAD OVER HEELS	A 20
1010-DAMAS	A 17
1011-SURVIVOR	A 20
1012-GAUNTLET	A 20
1013-DONKEY KONG	A 20
1014-PHANTOMAS II	A 20
1015-INTERNATIONAL KARATE	A 20
1016-KNOCK OUT 3D	A 20
1017-SPACE SHUTTLE	A 20
1018-DEMONIA	A 20
1019-MASTER OF THE LAMPS	A 20
1020-KONAMI'S BOXING	A 17
1021-007 THE LIVING DAYLIGHTS	A 20
1022-TURBO CHESS	A 20
1025-COSA NOSTRA	A 20
1026-AUF WIEDERSEHEN MONTY	A 20
1027-SPIRITS	A 20
1028-FEUD	A 20
1029-VESTRON	A 20
1030-AVENGER	A 20
1031-WINTER GAMES	A 20
1032-FERNANDO MARTIN BASKET	A 20
1033-WHO DARES WINS II	A 20
1036-THE LAST MISSION	A 20
1037-LIVINGSTONE SUPONGO	A 20
1038-MARTIANOIDS	A 20
1039-COLT 36	A 20
1040-BMX SIMULATOR	A 16
1041-VIDEO POKER	A 16
1042-DESOLATOR	A 16
1043-COLONY	A 16
1044-KRAKOUT	A 16
1045-ARMY MOVES	A 16
1046-HYPERSPORTS 3	A 14
1047-MOPIRANGER	A 14
1048-JET SET WILLY II	A 16
1049-CHORO Q	A 20
1050-PENTAGRAM	A 20
1051-NONAMED	A 20
1052- SUPER BOWL	A 20
1053-CYBERUM	A 20
1054-BEACH HEAD	A 20
1055-CITY CONNECTION	A 17
1056-SPIT FIRE 40	A 20
1057-THE HEIST	A 20
1058-TRAILBLAZER	A 20
1059-SEA KING	A 20
1060-SPY vs. SPY II	A 20
1061-SAMURAI NINJA III	A 20
1063-DUSTIN	A 20
1064-DEATH WISH III	A 20
1065-LEGEND OF KAGEE	A 20
1066-FUZZBALL	A 20
1067-10th FRAME	A 20
1068-DYNAMITE DAN	A 20
1069-MUTANTMONTY	A 20
1070-HOWARD THE DUCK	A 20
1071-COSMIC SHOCK ABSORVER	A 20
1072-THINGS BOUNCES BACK	A 20
1073-PHANTIS	A 20
1075-SCENTIPEDE	A 20
1076-ACE OF ACES	A 20
1077-POLICE ACADEMY	A 20
1078-TANK	A 20
1079-GODZILLA	A 20
1080-SABOTEUR MISSION	A 20
1081-KILLER TOMATOES	A 20
1082-ANTARES	A 20
1083-STOP BALL	A 20
1084-COBRA'S ARC	A 20
1085-ROCKY	A 20
1086-ALIENS "EL REGRESO"	A 20

UTILITARIOS

0829-DESENSAMBLADOR	A 53
0830-ENSAMBLADOR	A 53
0831-CONTABILIDAD GENERAL	A 17
0832-FICHEROS	A 17
0833-PASCAL	A 53
0834-MINILOGO	A 26
0835-PLANILLA DE CALCULO	A 17
0887-CONTROL DE STOCK	A 17
0911-CONTROL BANCARIO	A 17
0936-TASWORD II	A 26
0993-EDDY II	A 73
1034-KNIGHT COMMANDER	A 26
1035 KIT DE ALINEACION	A 43

TITULOS: ____________________

NOMBRE Y APELLIDO: ____________________

DIRECCION: __________ C.P. __________ LOCALIDAD: __________

FORMA DE PAGO: Cheque/giro Contrarrembolso PROVINCIA __________

Cheques y giros a la orden de Edmundo A. Goldin. Gastosde Envío A 16

RELOJ DIGITAL

Con este recurso podremos utilizar la computadora con precisión más que aceptable tanto para medir la duración de sucesos externos al sistema (ejemplo: la ejecución de una pieza musical), como para cronometrar determinados algoritmos que se encargan de tareas comúnmente lerdas, tales como el ordenamiento de vectores y búsquedas.

Una de las tantas ventajas que nos brinda la norma MSX es la de poseer un "reloj interno" que funciona automáticamente, sin intervención alguna del usuario.

MIDIENDO EL TIEMPO

Realmente el citado "reloj interno" es tan solo una variable fija del sistema llamado TIME, de tipo contador, que se incrementa 50 veces por segundo, una con cada ciclo de la corriente eléctrica. Esta variable es fácilmente accesible desde el BASIC;se puede realizar (poniendo su valor a CERO) en cualquier punto de un programa (TIME=0) o actuar sobre ella como con cualquier otra.

LIMITACIONES

Dado que la variable TIME solo puede almacenar valores enteros positivos (números naturales) hasta el 65535 (64 * 1024), tal como lo presentamos en los dos listados: DIGItron y INTERkron, solo pueden medirse lapsos de hasta algo más de veinte minutos (exactamente 21 min. 50 seg. 70/100), suficiente en la mayoría de los casos. Si se requiriera mayor duración, deberían efectuarse algunas modificaciones, tales como agregar una variable extra.

Otra limitación es la de la exactitud intrínseca del programa. Si se desea aumentar la misma, pueden modificarse algunas líneas del programa, agregando algunos REM o simplificando algunas sentencias para retardar o acelerar el proceso y ajustarlo al ritmo de un cronómetro patrón.

ALGORITMOS DE CALCULO

Dado que sabemos que TIME se incrementa en forma automática cincuenta veces por segundo, dividiendo el valor de la variable por 50, tendremos el tiempo transcurrido medido en segundos (ver figura 1).



Para descomponer esa cantidad en minutos y segundos, solo resta simplemente dividirla por 60. La parte entera del COCIENTE de esa operación indica los minutos, y el RESTO de la misma corresponde a los segundos restantes (ver figura 2).

Para calcular el RESTO de una división, se puede recurrir a utilizar una fórmula algo compleja: R = INT (T - 60 * INT (T / 60)), o a la función residente en el BASIC MSX que nos calcula este dato en forma directa, que es MOD. Obviamente utilizaremos la forma predefinida por su sencillez.

FIGURA 1

```
LET T = TIME / 50
```

FIGURA 2

```
TIME/50 : 60

  SEG        MIN
```

FIGURA 3

```
LET MIN = INT ( T / 60 )
LET SEG = T MOD 60
LET DC = (T - INT (T)) * 100
```

FIGURA 4

```
10 REM ******************
12 REM *                *
14 REM * D I G I t r o n *
16 REM *                *
18 REM *  Reloj digital  *
20 REM *                *
22 REM ******************
24 REM
26 REM por Gustavo Delfino
28 REM
30 OPEN "GRP:" AS #1
32 COLOR 1,10,10 : SCREEN 3
34 PSET (5,80)
36 PRINT #1,"DIGItron"
38 SOUND 1,0 : SOUND 8,0
40 SOUND 7,254 : SOUND 8,16
42 SOUND 13,1 : SOUND 0,60
44 SOUND 1,0 : SOUND 11,230
46 SOUND 12,35
48 IF INKEY$ = "" THEN GOTO 48
50 SOUND 13,1 : SOUND 0,90
52 SCREEN 0:KEY OFF:COLOR 4,15
54 PRINT "Pulse cualquier tecla
         para comenzar"
56 LOCATE 9,3:PRINT "┌────────┐"
58 LOCATE 9,4:PRINT "│        │"
60 LOCATE 9,5:PRINT "├────────┤"
62 LOCATE 9,6:PRINT "│        │"
64 LOCATE 9,7:PRINT "└────────┘"
66 LOCATE 10,4:PRINT "mm:ss:dc"
68 LOCATE 10,6:PRINT " 0: 0: 0"
70 IF INKEY$ = "" THEN GOTO 70
72 LOCATE,10 : PRINT "Pulse cual
   quier tecla para finalizar"
74 TIME = 0
76 REM - - - - - - - - - - - - -
78 LET T = TIME / 50
80 LOCATE 10,6 : PRINT USING "##
   :##:##" ; INT (T/60) ; T MOD
   60 ; ( T - INT (T)) * 100
82 IF INKEY$ = "" THEN GOTO 78
84 REM - - - - - - - - - - - - -
86 LOCATE 0,12
88 SOUND 13,1:SOUND 0,90
90 CLOSE # 1 : KEY ON
92 END
```

llez inherente.
Finalmente, para calcular las fracciones de segundo (centésimas), basta con restar de T su parte entera y al resultado multiplicarlo por 100 (ver figura 3).

DIGITRON: UN CRONOMETRO DE MESA

Este primer programa nos permite medir el tiempo transcurrido entre el instante en que pulsamos cualquier tecla por primera vez, para poner en funcionamiento el cronómetro (literalmente hablando) y la segunda, para detenerlo.
Resulta más que obvio precisar que la utilidad de este pequeño programa solo está limitada a la exploración de las posibilidades que nos brinda el BASIC de la norma MSX, más que al uso en sí que podamos llegar a darle. Cualquier reloj de pulsera cumple con esta misma finalidad con ventajas, salvo en cuanto a lapsos muy breves de unos pocos segundos, en los cuales esta propuesta puede ser más adecuada.
Obsérvese en el listado del programa, el uso que hemos hecho de: a) el modo de pantalla 3 (SCREEN 3) = GRAFICOS EN BAJA RESOLUCION para el título; b) el

FIGURA 5

```
10 REM ********************
14 REM *                  *
18 REM * I N T E R k r o n *
22 REM *                  *
26 REM ********************
30 REM
34 REM  por Gustavo Delfino
38 REM
42 REM - - - - - - - - - - - - - -
46 REM Ubique la l/nea siguiente
   al COMIENZO del acontecimien-
   to que desea cronometrar.
50 TIME = 0 : LET T1 = TIME / 50
54 REM - - - - - - - - - - - - - -
58 REM Ubique la l/nea siguiente
   al FINALIZAR el suceso.
62 LET T2 = TIME / 50
66 REM - - - - - - - - - - - - - -
70 LET T = T2 - T1
74 LET MIN = INT ( T / 60 )
78 LET SEG = T MOD 60
92 LET DC = (T - INT (T)) * 100
86 PRINT "Lapso transcurrido"
90 PRINT MIN;"m";SEG;"s";DC;"dc"
94 END
```

uso de los sonidos; c) el empleo de caracteres gráficos predefinidos para enmarcar el reloj y d) el manejo de pantalla con LOCATE y PRINT USING.
Nótese también que en el módulo operativo en sí, encerrado entre líneas con RAM, no se ha recurrido al almacenamiento previo de los valores resultantes del cálculo, sino que estos resultados son directamente exhibidos por medio del PRINT. Esto ha sido efectuado de esta manera, al solo efecto de ahorrar algunas fracciones de segundo en cada ciclo (ver figura 4).

INTERKRON: TOMANDO EL TIEMPO

Este segundo programa nos permite medir el tiempo que demora en ejecutarse una rutina de algún programa, que por su repetitividad pueda llegar a convertirse en el eslabón más lento del conjunto, tal como pueden ser las rutinas de ordenamiento de datos estructurados homogéneos (VECTORES).
Con este recurso de medición, podemos finalmente constatar cuántos segundos o fracciones de segundo ahorramos cuando eliminamos REMarks, LETs, indentado de ciclos u otros recursos que hacen a la programación más ordenada y de más fácil lectura e interpretación.
Sugiero someter a medición de tiempo las dos formas de un mismo programa, una con los REM que requiera, los LET, etcétera, y la otra quitándole los mismos (ver figura 5).

Gustavo O. Delfino

Sueldos MSX

Clase: Comercial
Autor: Sergio Segura

En esta versión reducida de Sueldos, ofrecemos no solo la emisión de dicho recibo, sino también la combinación de varias subrutinas interesantes...

Este programa realiza el cálculo, impresión y posterior grabación de una liquidación de haberes, en base a conceptos tales como:

Haberes con aporte:
1) Sueldo Bruto
2) Premios
3) Horas Extras
4) Sueldo Anual complementario

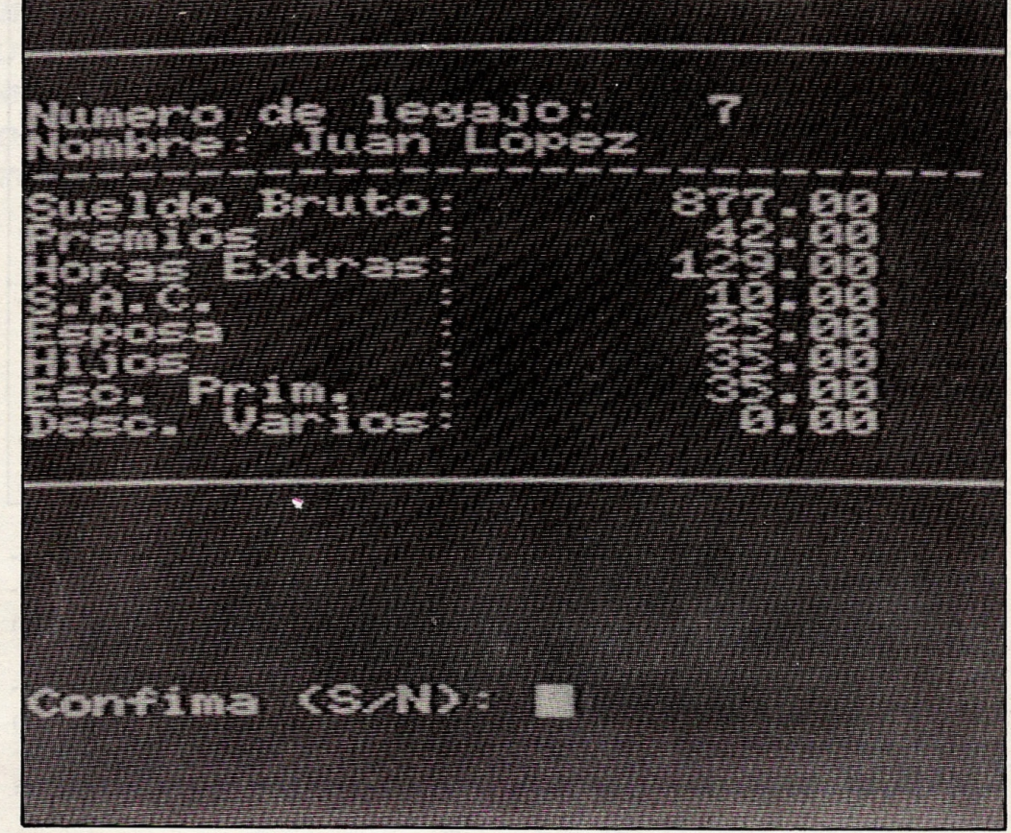
Haberes sin aportes:
5) Esposa
6) Hijos
7) Escolaridad

Retenciones:
8) Descuentos Varios
9) Jubilación 11%
10) Obra Social 3%
11) Ley 19.032 1%

Desde luego que se han elegido los conceptos básicos para una liquidación normal y también para el caso de aguinaldo. Esto no quita que el ingenio del lector haga susceptibles mejoras al mismo, ampliando la cantidad de conceptos, su cálculo y salida a impresora si lo desea. El mismo cuenta con algunas subrutinas que, gracias a su estructura modular, son de uso general para cualquier sistema. Ejemplo de ello es la rutina de control de ingreso de datos, Manejo de Campos en pantalla y Boxes. Estas funcionan con el envío de parámetros (Datos de Entrada y Salida).

Estructura del programa:

10-110: Apertura de archivos e incializaciones varias
120-400: Area de Datas
1000-1400: Control del Programa
2000-2900: Ingreso de Datos
3000-3400: Cálculo de liquidación
4000-4500: Impresión del recibo de Sueldos
5000-5100: Impresión de Datas en Pantalla
6000-6170: Lectura del Archivo
6200-6260: Grabación del Archivo
6300-6400: Impresión de los Campos
10000-10480: Dibuja un "marco" en el visor
30000-32180: Manejo de Campos en Pantalla
50000-50800: Errores



Variables Importantes:

L1$=Número de Legajo
N1$=Nombre del empleado
CA$()=Cantidades e importes a ingresar
RE()=Importes surgidos del cálculo
HCA#=Total de Haberes sujetos a descuentos
HSA#=Total de Haberes sin retención
DTS#=Retenciones a Realizar
CN$()=Estructura de archivo
ZZ$=Variable a ser modificada por la subrutina 30000
TI$=Tipo de Variable (A/N/F)
A$=Consistencia del dato
L=Longitud del campo
ND%=Número de decimales
X=Coordenada del cursor en Columna
Y=Coordenada del cursor en Fila
CL%=Longitud del "marco" en columnas
FLI%=Cantidad de filas que ocupará el "marco"

Cabe aclarar que para la liquidación de aguinaldo se deberá colocar donde dice "S.A.C.", el mejor sueldo percibido en el semestre; en el caso que este sea cero (0) se omitirá el cálculo del mismo, al igual que para el concepto Esposa, la cifra a colocar será 1,00 y en hijos la cantidad. Cuando los valores correspondientes a Subsidios deban ser modificados, se procederá a realizar su cambio en las líneas 3220-3260.

En el resto de los conceptos se ingresará directamente el importe a liquidar; todos ellos serán de signo positivo pues el sistema se encarga de realizar las sumas y restas correspondientes. Ej.: Descuentos Varios, se asume automáticamente como negativo. Lo mismo para Jubilación, Obra Social y Ley 19.032.

Finalmente, sugiero la ampliación de este "Sueldos en Escala Reducida", a través de la confección de programas adicionales tales como: Mantenimiento de Personal, Diversos listadores Oficiales (Planilla de Jubilación, Obra social, Sindicato, Libro Ley, etcétera), Estadísticos (Planilla de Cambio, Totales por Centro de Costo, Totales por concepto, Asiento Resumen...) y ,por supuesto, la emisión de recibo por impresora.

Una labor para pensar....

```
10 '==================================
20 ' Liquidacion de Sueldos
30 ' Para Msx
40 ' Programado por Sergio Segura
50 '==================================
60 '
65  CLEAR 5000
70  ON ERROR GOTO 50000
75  DIM CA$(20),CN$(20),RE(20)
80  OPEN "LIQUIDO.DAT" AS #1 LEN = 80
90  FIELD #1, 20 AS NO$, 4 AS CN$(1), 4  AS  CN$(2),  4  AS  CN$(3),  4  AS
CN$(4), 4 AS CN$(5), 4 AS CN$(6), 4 AS CN$(7), 4 AS CN$(8), 4 AS CN$(9),  4
AS CN$(10), 4 AS CN$(11)
100 CLS
110 KEY OFF
120 DATA 12,3,3,"Numero de legajo:          "
130 DATA   3,4 ,"Nombre:                    "
135 DATA   3,5 ,"---------------------------"
140 DATA   3,6 ,"Sueldo Bruto:              "
150 DATA   3,7 ,"Premios      :             "
160 DATA   3,8 ,"Horas Extras:              "
170 DATA   3,9 ,"S.A.C.       :             "
180 DATA   3,10,"Esposa       :             "
190 DATA   3,11,"Hijos        :             "
200 DATA   3,12,"Esc. Prim.  :              "
210 DATA   3,13,"Desc. Varios:              "
220 DATA   3,22,"Confima (S/N):   "
300 DATA 3,3,14,"Jubilac. 11%:              "
310 DATA   3,15,"Obra Soc. 3%:              "
320 DATA   3,16,"Ley 19032 1%:              "
400 '
1000 '--------------------------------
1010 '      Control del Programa
1020 '--------------------------------
1030 '
1050     L1$ = ""
1060     CLS
1100     X= 1 : Y = 1 : CL% = 35 :FL% = 15
1110     GOSUB 10000      'Dibuja Box
1120     RESTORE 120
1140     GOSUB 5000          'Imprime 1ra
1200     GOSUB 2000          'Ingreso
1220     GOSUB 3000          'Calcula
1240     GOSUB 4000          'Recibo
1270     GOSUB 6200          'Graba
1300 GOTO 1000
1400 '
2000 '---------------------------------
2010 '      Ingreso de Datos
2020 '---------------------------------
2030 '
2040 ZZ$ = L1$ : A$ ="" : TI$ = "N" : L = 3: ND% = 0 : X =21: Y=3
2050     GOSUB 30000
2060     L1$ = ZZ$
2070     IF VAL(L1$) = 0 THEN BEEP : GOTO 2040
2075     IF VAL(L1$) = 999 THEN CLS: END
2080     GOSUB 6100          'LECTURA
2085     GOSUB 6300          'IMPRESION
2090 '
2100 ZZ$ = N1$ : A$ ="" : TI$ = "A" : L = 20: ND% = 0 : X = 11: Y = 4
2120     GOSUB 30000
2140     N1$ = ZZ$
2190 '
2210 Y=6
2220 FOR AA=1 TO 8
2300 ZZ$ = CA$(AA) : A$ ="" : TI$ = "N" : L = 9: ND% = 2 : X = 18
2320     GOSUB 30000
2340     CA$(AA) = ZZ$
2350     Y = Y +1
2360 NEXT AA
2390 '
2400 ZZ$ = OP$ : A$ = SN" : TI$ = "A" : L = 1: ND% = 0 : X = 18 : Y = 22
2420   GOSUB 30000
2440   OP$ = ZZ$
2450   IF OP$ = " " THEN 2400
2460   IF OP$ = "N" THEN 2100
2470   IF OP$ = "F" THEN 1000
2490 '
2900 RETURN
2910 '
3000 '--------------------------------
3010 '        CALCULO
3020 '--------------------------------
3030 '
3100   HCA# = 0
3110   FOR AA = 1 TO 3
3120       RE(AA) = VAL(CA$(AA))
3130       HCA# = RE(AA) + HCA#
3140   NEXT AA
3160   IF HCA# <> 0 THEN RE(4) = VAL(CA$(4)) /2 : HCA# = HCA# + RE(4)
3190 '
3220   IF VAL(CA$(5)) <>0 THEN RE(5) = 15
3240   RE(6) = VAL(CA$(6)) * 15
3260   RE(7) = VAL(CA$(7)) * 20
3270   HSA# = RE(5) + RE(6) + RE(7)
3290 '
3310   RE(8) = - VAL(CA$(8))
3320   RE(9) = HCA# * - .11     'Jubil
3340   RE(10) = HCA# * - .03    'Obra Soc
3360   RE(11)= HCA# * - .01    'Ley 19032
3380   DTS# = RE(8) + RE(9) + RE(10) + RE(11)
3390 '
3400 RETURN
3410 '
4000 '--------------------------------
4010 '  Imprime Recibo por Pantalla
4020 '--------------------------------
4030 '
4040 CLS
4100   X = 1 : Y = 1 : CL% = 35 : FL% =17
4110   GOSUB 10000       'Dibuja Box
4120   RESTORE 120
4130   GOSUB 5000        'Imprime 1ra
4140   RESTORE 300
4150   GOSUB 5000        'Imprime 2da
4160   LOCATE 22,3 : PRINT USING "###";VAL(L1$)
4170   LOCATE 12,4 : PRINT N1$
4210   Y = 6
4220   FOR AA =1 TO 11
4230       LOCATE 18,Y
4240       PRINT USIN$"#########,.##";RE(AA)
4250       Y = Y + 1
4260   NEXT AA
4300   LOCATE 3,18 : PRINT USING"Tot. Haberes:       ######.##"; HCA#
4310   LOCATE 3,19 : PRINT USING"Tot. Subsid.:       ######.##"; HSA#
4320   LOCATE 3,20 : PRINT USING"Tot. Desc.  :       ######.##"; DTS#
4330   LOCATE 3,21 :  PRINT  USING"NETO  A  COBRAR:       ######.##  *---";
HCA#+HSA#+DTS#
4400 ZZ$ = OP$ : A$ ="SN" : TI$ = "A" : L = 1: ND% = 0 : X = 18 : Y = 22
4410 '
4420   GOSUB 30000
4440   OP$ = ZZ$
4450   IF OP$ = " " THEN 4400
4460   IF OP$ = "N" THEN 1000
4500 RETURN
5000 '--------------------------------
```

```
5010 '    Dibuja Pantalla de datos
5020 '-----------------------------------
5030 '
5050    READ B
5060    FOR AA = 1 TO B
5070       READ X,Y,SS$
5080       LOCATE X,Y,O : PRINT SS$
5090    NEXT AA
5100    RETURN
5200 '
6000 '--------------------------------
6010 ' Lectura, Escritura e Impresion
6020 '--------------------------------
6030 '
6100 GET #1,VAL(L1$)
6120    N1$ = NO$
6140    FOR AA = 1 TO 11
6150       CA$(AA) = STR$(CVS(CN$(AA)))
6160    NEXT AA
6170 RETURN
6200 '
6210    LSET NO$ = N1$
6220    FOR AA = 1 TO 11
6230       LSET CN$(AA) = MKS$(VAL(CA$(AA)))
6240    NEXT AA
6250 PUT #1,VAL(L1$)
6260 RETURN
6300 '
6310 LOCATE 11,4 : PRINT N1$
6320 FOR AA = 1 TO 8
6340    LOCATE 18,AA + 5
6350    PRINT USING"#######.##";VAL(CA$(AA))
6380 NEXT AA
6390 RETURN
6400 '
10000 ' -----------------------------
10010 '          BOXES
10020 ' -----------------------------
10030 '
10120    LOCATE X,Y,O
10130    PRINT CHR$(1) + CHR$(88);
10140    FOR AA = 1 TO CL%-2
10150       PRINT CHR$(1) + CHR$(87);
10160    NEXT AA
10170    PRINT CHR$(1) + CHR$(89);
10180    FOR AA = Y+1 TO Y+FL%-2
10210       LOCATE X,AA,O : PRINT CHR$(1)+CHR$(86)
10240       LOCATE X+CL%-1,AA,O : PRINT CHR$(1)+CHR$(86)
10270    NEXT AA
10420    LOCATE X,AA,O
10430    PRINT CHR$(1) + CHR$(90);
10440    FOR AA = 1 TO CL%-2
10450       PRINT CHR$(1) + CHR$(87);
10460    NEXT AA
10470    PRINT CHR$(1) + CHR$(91)
10480 RETURN
10490 '
30000 '**************************************
30010 '*  MANEJO DE CAMPOS EN PANTALLA      *
30020 '**************************************
30030 '
30035 IF ZZ$ = "" THEN ZZ$ = STRING$(L%," ")
30040 IF TI$="N" OR TI$="n" THEN 31000
30050 IF LEN(ZZ$)< L THEN ZZ$ = ZZ$ + SPACE$(L- LEN(ZZ$))
30060 '
30070 '********* alfanumericos ********
30080 '
30090 A=1
30100 LOCATE X,Y,O:PRINT ZZ$:LOCATE X+A-1,Y,1:B$=INPUT$(1)
30110 IF ((A$="" AND ASC(B$) > 31 AND ASC(B$) <> 127) OR (INSTR(1,A$,B$) <>
O)) AND A<=L THEN LOCATE X,Y,O:MID$(ZZ$,A,1)=B$:A=A+1
30120 IF ASC(B$)=127 AND A<L+1 THEN ZZ$=MID$(ZZ$,1,A-1)+MID$(ZZ$,A+1,L-A)+"
"
30130 IF ASC(B$)=8  AND  A>1  THEN  ZZ$=MID$(ZZ$,1,A-2)+MID$(ZZ$,A,L-A+1)+"
":A=A-1
30140 IF ASC(B$)=18 AND A<L THEN ZZ$=MID$(ZZ$,1,A-1)+" "+MID$(ZZ$,A,L-A)
30150 IF ASC(B$)=21  THEN 30080
30155 IF ASC(B$)=27  THEN ZZ$=STRING$(L,32) :A=1
30160 IF ASC(B$)=28 AND A<L+1 THEN A=A+1
30170 IF ASC(B$)=29 AND A>1 THEN A=A-1
30180 IF ASC(B$) <>13 THEN 30100
30190 IF TI$ = "f" OR TI$="F" THEN 32000
30200 Q=INSTR(1,ZZ$," "):IF Q=0 THEN  LOCATE  X,Y,O:PRINT  ZZ$:RETURN  ELSE
MID$(ZZ$,Q,1)="":RETURN
30210 '
31000 '********** NUMERICOS *********
31010 '
31020 IF VAL(ZZ$) <>0 THEN ZZ$= STR$(VAL(ZZ$)*10^ND%)
31025 IF LEN(ZZ$)< L THEN ZZ$ = SPACE$(L- LEN(ZZ$)) + ZZ$
31030 IF ND%=0 THEN FO$=STRING$(L,"#") : GOTO 31050
31040 FO$=STRING$(L-ND%,"#")+"."+STRING$(ND%,"#")
31050 A=1
31060    LOCATE     X,Y,O:PRINT     USING     FO$;VAL(ZZ$)/10^ND%:LOCATE
X+L,Y,1:B$=INPUT$(1)
31070 IF ((A$="" AND ASC(B$) >= 48 AND ASC(B$) =< 57 )  OR  (INSTR(1,A$,B$)
<> O)) AND A<=L THEN LOCATE X,Y,O:MID$(ZZ$,A,1)=B$:A=A+1:
31080 IF ASC(B$)=8 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31090 IF ASC(B$)=127 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31100 IF ASC(B$)=27 THEN ZZ$=STRING$(L," ") : GOTO 31050
31110 IF ASC(B$)=28 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31120 IF ASC(B$)=29 AND A<=L THEN MID$(ZZ$,A,1)="O" :A=A+1 :
31130 IF ASC(B$) <>13 THEN 31060
31135 ZZ$ =STR$(VAL(ZZ$)/10^ND%)
31140   Q=INSTR(1,ZZ$," "):IF  Q=O  THEN  LOCATE  X,Y,O:PRINT  USING
FO$;VAL(ZZ$)/100^ND%:RETURN ELSE MID$(ZZ$,Q,1)="":RETURN
31150 '
31180 '
32000 '*********** fechas ************
32010 '
32020   IF SALI% = 1 THEN 32070 ELSE SALI% = 1
32030 DIM DS(12)
32040 DATA 31,28,31,30,31,30,31,31,30,31,30,31
32050 RESTORE 32040
32060 FOR AA=1 TO 12 : READ DS(AA) :NEXT
32070 AN$=MID$(ZZ$,5,2):AN=VAL(AN$)
32080 IF AN<86 THEN BEEP : GOTO 30000
32090 '
32100 ME$=MID$(ZZ$,3,2):ME=VAL(ME$)
32110 IF ME <1 OR ME >12 THEN BEEP :GOTO 30000
32120 '
32130 DI$=MID$(ZZ$,1,2) :DI=VAL(DI$)
32140 IF (AN MOD 4 =0) AND  (ME = 2) THEN DS(2)=29
32150 IF DI<1 OR DI>DS(ME) THEN BEEP : GOTO 30000
32160 ZZ$=DI$+ME$+AN$
32180 RETURN
32190 '
50000 ' ------------------------------
50010 '          Errores
50020 ' ------------------------------
50030 '
50040  IF ERR = 55 AND ERL = 6100 THEN RESUME 6170
50060  LOCATE 3,22 : PRINT USING"Error ## en linea ##### ";ERR;ERL
50070  A$ = INPUT$(1)
50080  CLOSE : END
```

CONCURSO MENSUAL

PREMIO

MSX-TEST



Un soft a eleccion entre IDEA BASE, IDEA TEXT, o BASIC TUTOR.

CONCURSO MENSUAL

Auspiciado por TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Para participar en este certamen deben señalar cual es la información correcta que presenta cada ítem. Junto con las respuestas deben remitir los datos en el correspondiente cupón. Los premios se entregarán en la administración de la editorial. Los que no puedan concurrir pueden solicitar el envio pagando el franqueo contrareembolso. Los premios podrán ser reclamados dentro de los 120 días después de haber sido anunciados.

Ganador del MSX-TEST Nº 1

Rubén P. Papazian Pcia. de Córdoba

Respuestas correctas:

1-área de hooks. 2-El lenguaje Basic de IBM. 3-Indica cuantos caracteres contiene una expresión alfanumérica. 4-9. 5-Prueba de escritorio.

MSX-TEST Nº 5 - CIERRE 1º DE JULIO DE 1988

1. ¿Cuál es el material básico para construir chips?
a. aluminio
b. cobre
c. silicio

2. ¿Qué es FAT?
a. Tabla de ubicación de archivos
b. Tiempo de acceso de archivo
c. Sector principal del archivo

3. ¿Cuál de estos tres pares de sentencias no conforman un ciclo?
a. WHILE/DO
b. REPEAT/UNTIL
c. WHEN/NEXT

4. ¿Cuándo se ejecuta más rápido un programa?
a. Está compilado
b. Es interpretado
c. Las opciones a) y b) tardan lo mismo.

5. ¿Cómo se llaman los Puestos de Atención Servicio ACAMATICA?
a. SERVIACA
b. PASA
c. TELEACA

Nombre y apellido____________________

Dirección:____________________

Documento:__________ Edad:_____ Máquina:__________

Qué es lo que más me gusta de la revista:____________________

Que le agregaría:____________________

Que es lo que no me gusta:____________________

EXPECTATIVA DE VIDA

Clase: Educativo

Con este programa, basado en la experiencia científica sobre el tema, podremos calcular hasta qué edad habitaremos este suelo.



Este programa no pretende darle ilusiones de inmortalidad ni tampoco amargarle la vida al usuario, pero por lo menos nos hace reflexionar sobre las causas probables de mortalidad y así tratar de poner freno a las tensiones cotidianas.

Está basado en un estudio estadístico con miles de casos analizados y logra el propósito de entretener, además de ser didáctico.

Su uso es muy sencillo. Una vez cargado, coloquemos el teclado para ser usado en mayúsculas, y luego es cuestión de responder a las diferentes preguntas que nos efectúa el programa.

Sobre la base de nuestra edad actual y el sexo se calcula una edad base de la que se empiezan a sumar o restar años de acuerdo con las contestaciones a las siguientes preguntas.

El formulario está dividido en cuatro partes. La primera tiene que ver con la herencia (enfermedades de nuestros antepasados, causas de muerte, etcétera).

En la segunda las preguntas se refieren a nuestra dieta y alimentación. Aquí nos puede sorprender alguna variación no imaginada del resultado parcial. Todo tiene su explicación, y el programa la dará si es preciso.

La parte tres se refiere a la educación y ocupación, mientras que la última está dedicada a nuestro estilo de vida.

Una aclaración: el programa está hecho para personas mayores de veinte años, no por capricho sino que esa es la base científica tomada.

Esperamos que sean sinceros al responder las preguntas y tengan larga vida.

```
5 WIDTH 37:KEYOFF
10 CLS : PRINT TAB(10) "<<
LIFETIME >>"
20 PRINT: PRINT "TRABE LAS
MAYUSCULAS": PRINT: FOR I=1
TO 200:NEXT I
30 PRINT :PRINT "ESTE PRO-
GRAMA DETERMINA SU EXPECTATI-
VA DE VIDA"
40 PRINT :PRINT "UD. DESEA MAS
INFORMACION";
50 LINE INPUT "? ";A$
60 IF LEFT$(A$,1)="N" THEN
150
70 PRINT
80 PRINT "LA SIGUIENTE EX-
POSICION ESTA BASADA "
90 PRINT "SOBRE LA MEJOR
EVIDENCIA CIENTIFICA "
95 PRINT "DEL MOMENTO."
98 PRINT
100 PRINT "MIENTRAS LOS CIEN-
TIFICOS NO HAN DES- CUBIERTO
TODAS LAS VARIABLES QUE
 PRO-";
110 PRINT "LONGAN LA VIDA,
ELLOS SE HAN ACERCADO A
ALGUNOS DE LOS FENOMENOS QUE
ESTAN";
120 PRINT "RELACIONADOS CON LA
LONGEVIDAD."
125 PRINT
130 PRINT "ESTE PROGRAMA ESTA
BASADO SOBRE ESOS DATOS Y
COMIENZA CON INDIVIDUOS
A "
140 PRINT "PARTIR DE LOS 20 A%OS
DE EDAD."
150 DIM B1(101)
160 PRINT :PRINT "CUAL ES SU
EDAD EN A%OS";
170 INPUT A9
180 IF A9>19 THEN GOTO 190 ELSE
PRINT "LO LAMENTO, UD. DEBERA
CONTAR CON MAS DE
20 A%OS DE EDAD." :
PRINT"PRESIONE UNA TECLA"
185 A$= INPUT$(1) : CLOSE : RUN
190 PRINT :PRINT "CUAL ES SU
SEXO";
200 LINE INPUT "? ";S$
210 FOR I=0 TO 101
220 READ B1(I)
230 NEXT I
240 C=A9-20
250 IF LEFT$(S$,1)="F" THEN
C=C+51
260 DATA 69.9, 70.0, 70.1,
70.2, 70.3, 70.4,70.5, 70.6,
70.7, 70.8
270 DATA 70.9, 70.9, 71.0
,71.1, 71.2, 71.3, 71.4,
71.5, 71.6, 71.7
280 DATA 71.8, 71.0, 72.0,
72.2, 72.3, 72.5, 72.6, 72.8,
73.0, 73.2
290 DATA 73.4, 73.6, 73.8,
74.1, 74.3, 74.6, 74.9, 75.2,
75.5, 75.8
300 DATA 76.2, 76.5, 76.9,
77.3, 77.7, 78.1, 78.6, 79.0,
79.5, 79.9
310 DATA 80.4
320 DATA 77.0, 77.0, 77.1,
77.1, 77.2, 77.2, 77.3, 77.3,
77.4
330 DATA 77.4, 77.5, 77.5,
77.5, 77.6, 77.6, 77.7, 77.8,
77.8, 77.9
340 DATA 78.0, 78.0, 78.1,
78.2, 78.3, 78.4, 78.5, 78.6,
78.7, 78.9
350 DATA 79.0, 79.1, 79.3,
79.4, 79.6, 79.7, 79.9, 80.1,
80.3, 80.5
360 DATA 80.7, 80.9, 81.2,
81.4, 81.6, 81.9, 82.2, 82.4,
```

```
82.7, 83.0
370 DATA 83.3,83.6
380 PRINT :PRINT "SU EXPECTA-
TIVA BASE DE VIDA ES
";STR$(B1(C));" A%OS"
390 PRINT :T=B1(C)
395 FOR AA= 1 TO 1000 : NEXT
AA
400 CLS
405 GOSUB 2680
410 PRINT "PARTE I - HEREN-
CIA"
420 GOSUB 2680
430 PRINT :PRINT "TUVO DOS
ANTEPASADOS QUE HAYAN
VIVIDOHASTA LA EDAD DE 80 O
MAS
";
450 LINE INPUT "? ";A$
460 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 0
=";T:GOTO 490
470 PRINT TAB(10);T;"+ 2
=";T+2
480 T=T+2
490 PRINT
500 PRINT "CUAL ES/FUE LA
EDAD DE SU MADRE";
510 INPUT A
520 IF A<80 THEN PRINT
TAB(10);T;"+ 0 =";T:GOTO 540
530 PRINT TAB(10);T;"+ 1.5
=";T+1.5:T=T+1.5
540 PRINT
550 PRINT "CUAL ES/FUE LA
EDAD DE SU PADRE";
560 INPUT A
570 IF A<80 THEN PRINT
TAB(10);T;"+ 0 =";T:GOTO 590
580 PRINT TAB(10);T;"+ 2
=";T+2:T=T+2
590 PRINT
600 PRINT "TUVO ALGUN
ANTEPASADO,PARIENTE,HERMA-
";
610 PRINT "NO/A, QUE HAYA
FALLECIDO DE UN ATAQUEAL
CORAZON O PRESION ARTERIAL AN
TES "
620 PRINT "DE LOS 50 A%OS";
630 LINE INPUT "? ";A$
640 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"- 0
=";T:GOTO 670
650 PRINT TAB(10);T;"- 4
=";T-4
660 T=T-4
670 PRINT
680 PRINT "TUVO ALGUN PARI-
ENTE MUERTO, DE LOS YAMEN-
CIONADOS, POR LAS MISMAS CAUS
AS"
690 PRINT "ANTES DE LOS 60
A%OS";
700 LINE INPUT "? ";A$
710 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"- 0
=";T:GOTO 740
720 PRINT TAB(10);T;"- 2 =";T-
2
730 T=T-2
740 PRINT
750 PRINT "CUAL FUE EL NUMERO
TOTAL DE INCIDEN-"
760 PRINT "TES DE DIABETES,
DESORDENES EN TIROI-
ROIDES,CANCER DE MAMAS
(MUJERES)
,"
770 PRINT "CANCER EN SISTEMA
DIGESTIVO, ASMA, Y"
780 PRINT "BRONQUITIS CRONICA
ENCONTRADA EN PA- RIENTES Y
ANTEPASADOS"
790 INPUT A
800 PRINT TAB(10);T;"- ( 3
X";A;") =";T-3*A
810 T=T-3*A:PRINT
815 FOR AA= 1 TO 1000 : NEXT AA
818 CLS
820 GOSUB 2680
830 PRINT "PARTE II - DIETA Y
ALIMENTACION"
840 GOSUB 2680
850 PRINT
860 PRINT "CUANTOS 'KILITOS' DE
MAS POSEE";
870 INPUT A
880 IF A=0 THEN B=0:GOTO 900
890 B=A/10
900 PRINT TAB(10);T;"-
";B;"=";T-B
910 T=T-B
920 PRINT
930 PRINT "CUANTOS PAQUETES DE
CIGARRILLOS FUMA DIARIAMENTE";
940 INPUT A
950 IF A=0 THEN B=0:GOTO 1030
960 IF A<1 THEN B=2:GOTO 1030
970 IF A=1 THEN B=3:GOTO 1030
980 IF A<=1.5 THEN B=5.5:GOTO
1030
990 IF A<2 THEN B=7:GOTO 1030
1000 IF A=2 THEN B=8:GOTO 1030
1010 IF A<3 THEN B=10:GOTO 1030
1020 B=12
1030 PRINT TAB(10);T;"-
";B;"=";T-B
1040 T=T-B
1050 PRINT
1060 PRINT "CUANTA BEBIDA ALCO-
HOLICA CONSUME POR DIA (EN
LITROS)";
1070 INPUT A
1080 IF A=0 THEN PRINT
TAB(10);T;"- 1 =";T-1:T=T-
1:GOTO 1110
1090 IF A<=2 THEN PRINT
TAB(10);T;"+ 3
=";T+3:T=T+3:GOTO 1110
1100 PRINT TAB(10);T;"- 8 =";T-
8:T=T-8
1110 PRINT :E$="UD. DESEA UNA
EXPLICACION DE ESTO":PRINT E$;
1120 LINE INPUT "? ";A$
1130 IF LEFT$(A$,1)="N" THEN
1200
1140 PRINT
1150 PRINT "BEBIENDO MOD-
ERADAMENTE (HASTA DOS VASOS
POR DIA) REDUCE EL STRESS
 Y LA MALA DIGESTION."
1160 PRINT
1170 PRINT "BEBER EN EXCESO
PRODUCE DANOS FISIO-";
1190 PRINT "LOGICOS, DEPRE-
SION, ETC."
1200 PRINT
1210 PRINT "UD. REALIZA EJER-
CICIOS MODERADAMENTE
(AEROBISMO,GOLF, LARGAS
CAMINAT
AS, NADAR) 2 O 3 VECES POR
SEMANA";
1230 LINE INPUT "? ";A$
1240 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 0
=";T:GOTO 1260
1250 PRINT TAB(10);T;"+ 3
=";T+3:T=T+3
1260 PRINT
1270 PRINT "CUANTAS HORAS
DUERME POR DIA";
1280 INPUT "? ";A
1290 IF A<9 THEN PRINT
TAB(10);T;"- 0 =";T:GOTO
1320
1300 IF A=9 THEN PRINT
TAB(10);T;"- 4 =";T-4:T=T-
4:GOTO 1320
1310 PRINT TAB(10);T;"- 6
=";T-6:T=T-6
1320 PRINT :PRINT E$;
1330 LINE INPUT "? ";A$
1340 IF LEFT$(A$,1)="N" THEN
1390
1350 PRINT :PRINT "ADULTOS
QUE DUERMEN MUCHO, USAN DEMA-
SIADAS HORAS EN ACTIVIDA
DES NO FISI-"
1360 PRINT "CAS, SON INFE-
LICES Y DUERMEN COMO ES-CAPE
DE ELLO."
1370 PRINT "LAS PERSONAS
DEPRESIVAS TIENEN EXPEC-TA-
TIVA DE VIDA MAS CORTA."
1390 PRINT
1400 IF LEFT$(S$,1)="F" AND
A9<=30 OR LEFT$(S$,1)="M"
AND A9<=40 THEN 1550
1410 IF LEFT$(S$,1)="M" THEN
1490
1420 PRINT "UD. REALIZA EX-
AMENES DE MAMAS Y PAPA-NICO-
LAU (MUJERES), POR LO MENOS
UNA VEZ AL ANO"
1440 LINE INPUT "? ";A$
1450 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 0
=";T:GOTO 1470
1460 PRINT TAB(10);T;"+ 2
=";T+2:T=T+2
1540 PRINT
1550 FOR AA=1 TO 1000 : NEXT
```

```
AA
1554 CLS
1558 GOSUB 2680
1560 PRINT "PARTE III - EDU-
CACION Y OCUPACION"
1570 GOSUB 2680
1580 PRINT
1590 PRINT "COMENZANDO DESDE
PRIMER GRADO INFE- RIOR, CUAL
ES EL NUMERO DE ANOS
 QUE HA PERMANECIDO EN LA
ESCUELA";
1610 INPUT "? ";A
1620 IF A<12 THEN PRINT
TAB(10);T;"- 2 =";T-2:T=T-
2:GOTO 1660
1630 IF A<14 THEN PRINT
TAB(10);T;"+ 0 =";T:GOTO 1660
1640 IF A<17 THEN PRINT
TAB(10);T;"+ 1
=";T+1:T=T+1:GOTO 1660
1650 PRINT TAB(10);T;"+ 3
=";T+3:T=T+3
1660 PRINT :PRINT E$;
1670 LINE INPUT "? ";A$
1680 IF LEFT$(A$,1)="N" THEN
1730
1690 PRINT :PRINT "MAYOR CAN-
TIDAD ANOS CURSADOS NO HAR-
ANSU VIDA MAS LARGA, PERO
MAYOR EDUCA- CION ESTA RELA-
CIONADO ";
1710 PRINT "CON MAYORES
OPORTUNIDADES DE INCREMENTAR
SUS IN- GRESOS Y ACCEDER
 A UN MEJOR NIVEL DE VIDA"
1730 PRINT
1740 PRINT "SU OCUPACION ES
CALIFICADA COMO PRO-
FESIONAL";
1750 LINE INPUT "?";A$
1760 IF LEFT$(A$,1)="N" THEN
1810
1770 PRINT "ES UD. MUSICO,
ARQUITECTO O FARMACEU-TICO";
1780 LINE INPUT "? ";A$
1790 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 2
=";T+2:T=T+2:GOTO 1870
1800 PRINT TAB(10);T;"- 1
=";T-1:T=T-1:GOTO 1870
1810 PRINT "UD. TRABAJA EN
TAREAS PESADAS";
1820 'PRINT "COCINERO U OB-
RERO"
1830 LINE INPUT "? ";A$
1840 IF LEFT$(A$,1)="N" THEN
1860
1850 PRINT TAB(10);T;"- 2
=";T-2:T=T-2:GOTO 1870
1860 PRINT TAB(10);T;"+ 0
=";T
1870 PRINT :PRINT E$;
1880 LINE INPUT "? ";A$
1890 IF LEFT$(A$,1)="N" THEN
1970
1900 PRINT :PRINT "PROFESION-
ALES USUALMENTE VIVEN MAS
ANOS, EXCEPTO ARQUITECTO
S Y FARMACEU-TICOS. "
1920 PRINT "UD. TIENE UNA MAYOR
POSIBILIDAD DE ACCIDENTARSE
SI REALIZA TAREAS
RIES- GOSAS O PESADAS."
1940 PRINT "PERSONAS QUE ESTAN
ASOCIADAS A TARE- AS DE ALIMEN-
TACION TIENEN UNA M
ENOR EXPECTATIVA DE VIDA."
1970 PRINT
1980 PRINT "ES SU INGRESO ANUAL
SUPERIOR A         A 40.000";
1990 LINE INPUT "? ";A$
2000 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"- 0 =";T:GOTO
2020
2010 PRINT TAB(10);T;"- 2 =";T-
2:T=T-2
2020 PRINT :PRINT E$;
2030 LINE INPUT "? ";A$
2040 IF LEFT$(A$,1)="N" THEN
2070
2050 PRINT :PRINT "PERSONAS CON
MAYOR INGRESO USUALMENTEEX-
PERIMENTAN STRESS Y CO
NSUMEN MAS SEDANTES."
2070 PRINT
2080 PRINT "SU TRABAJO ES AC-
TIVO O SEDENTARIO";
2090 LINE INPUT "? ";A$
2100 IF LEFT$(A$,1)="A" THEN
PRINT TAB(10);T;"+ 3
=";T+3:T=T+3:GOTO 2120
2110 PRINT TAB(10);T;"- 3 =";T-
3:T=T-3
2120 PRINT
2130 IF A9<61 THEN 2180
2140 LINE INPUT "SE ENCUENTRA
UD. TRABAJANDO ";A$
2150 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 0 =";T:GOTO
2170
2160 PRINT TAB(10);T;"+ 2
=";T+2:T=T+2
2170 PRINT
2180 FOR AA=1 TO 1000 : NEXT :
CLS
2185 GOSUB 2680
2190 PRINT "PARTE IV - ESTILO DE
VIDA"
2200 GOSUB 2680
2210 PRINT
2220 PRINT "UD. VIVE EN UN AREA
RURAL O URBANA";
2230 LINE INPUT "? ";A$
2240 IF LEFT$(A$,1)="R" THEN
PRINT TAB(10);T;"+ 1
=";T+1:T=T+1:GOTO 2260
2250 PRINT TAB(10);T;"- 1 =";T-
1:T=T-1
2260 PRINT
2270 PRINT "ESTA UD. CASADO O
VIVIENDO CON AL- GUIEN SOBRE
UNA BASE ESTABLE"
2290 LINE INPUT "? ";A$
2300 IF LEFT$(A$,1)="S" THEN
PRINT TAB(10);T;"+ 3
=";T+3:T=T+3:GOTO 2370
2310 A=A9-25
2320 IF A<=0 THEN A=0:GOTO
2360
2330 PRINT "CUANTOS DE LOS
PASADOS";STR$(A);" A%OS PASO
EN SOLEDAD";
2340 INPUT "? ";A
2350 A=A/10
2360 PRINT TAB(10);T;"-
";A;"=";T-A:T=T-A
2370 PRINT
2380 PRINT "EN UNA ESCALA DE
0 A 3, QUE GRADO DE CALMA O
PASIVO SE CONSIDERA UD.
"
2390 PRINT "(3 ES EL MAS
CALMO)";
2400 INPUT A
2410 PRINT
TAB(10);T;"+";A;"=";T+A:T=T+A
2420 PRINT
2430 PRINT "EN UNA ESCALA DE
0 A 5, CUANTO DE A- GRESIVO,
INTENSO Y COMPETITIVO
SE CONSIDERA UD ( 5 ES EL ";
2450 PRINT "MAS AGRESIVO, IN-
TENSO, Y/O COMPETITIVO)";
2460 INPUT A
2470 PRINT TAB(10);T;"-
";A;"=";T-A:T=T-A
2480 PRINT
2490 PRINT "CONDUCE SU AUTO
USANDO CINTURONES DE SEGURI-
DAD Y RESPETA LOS LIMITES
 DE VELOCIDAD";
2510 LINE INPUT "? ";A$
2520 IF LEFT$(A$,1)="N" THEN
PRINT TAB(10);T;"+ 0
=";T:GOTO 2540
2530 PRINT TAB(10);T;"+ 1
=";T+1:T=T+1
2540 PRINT
2550 PRINT "SOBRE LA
SIGUIENTE ESCALA,COMO SE
CONSIDERA"
2560 PRINT
2570 PRINT "   FELIZ: 3 2 1
0 -1 -2 -3 :INFELIZ"
2580 PRINT TAB(12);
2590 INPUT A
2600 IF A>=0 THEN PRINT
TAB(10);T;"+";A;"=";T+A:T=T+A:GOTO
2620
2610 PRINT TAB(10);T;"-";(-
A);"=";T+A:T=T+A
2620 PRINT :PRINT
2630 GOSUB 2680
2640 PRINT TAB(8);"SU EXPEC-
TATIVA DE VIDA ES:
";STR$(T);" A%OS.";TAB(58)
2650 GOSUB 2680
2655 A$ = INPUT$(1)
2658 CLOSE : RUN
2660 PRINT :PRINT :PRINT
2680 PRINT STRING$(37,"-")
;:RETURN
```

PASAJE DE UNIDADES

Clase: Educativo
Autor: Humberto González

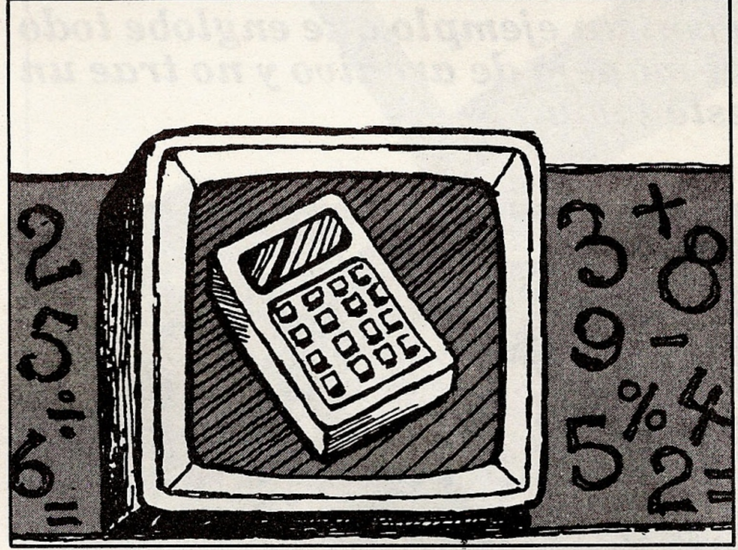
Este programa presenta las opciones para resolver en pocas líneas el tema del pasaje de unidades. El alumno trata de resolverlo y, en caso de que conteste mal, se le presentará una ayuda, explicando qué es lo que debe hacer para llegar al resultado correcto.

Con este programa, útil para la escuela primaria, mostramos cómo sacar provecho de las decisiones múltiples.

En muchas oportunidades necesitamos romper la secuencia de un programa para pasar a otro tramo del mismo.

Con el comando GO TO lo podemos lograr, aún cuando no existe condición alguna para que se produzca el salto deseado. Luego del GO TO debemos poner el número de línea al que queremos pasar.

Si existe alguna condición para pasar de un tramo a otro del programa se puede utilizar la instrucción IF...THEN..., que nos brinda la posibilidad de tener dos salidas: una si la condición es verdadera y la otra si es falsa.



Pero ocurre que muchas veces necesitamos más de dos salidas en una decisión y solemos caer en el error de acumular IF cuando tenemos una posibilidad mucho más rica a nuestro alcance.

Se trata de la decisión o bifurcación múltiple. La misma tiene esta forma:

ON (expresión) GOTO n1,n2,...,nn

donde los diferentes "n" son los números de línea a los que el programa debe ir si la expresión vale 1,2, etcétera.

La expresión puede ser una única variable o una expresión aritmética (A/B), (A+2*C), etcétera. Si el resultado es fraccionario, se toma la parte entera.

Dependiendo del valor que toma la expresión el programa irá a uno de los tantos números de línea. Por ejemplo, en el siguiente programa, si A vale 1 pasará a la línea 30, si vale 2 a la 40, si vale 3 a la 30 y si vale 4 a la 50.

```
10 INPUT A
20 ON A GOTO
30,40,30,50
30 B=A*2: GOTO 60
40 B=A/2: GOTO 60
50 B=A+2
60 PRINT B
```

El programa que ilustra esta nota utiliza decisiones múltiples para resolver en pocas líneas el tema de pasaje de unidades. Analizándolo veremos una de las formas de utilización de la instrucción

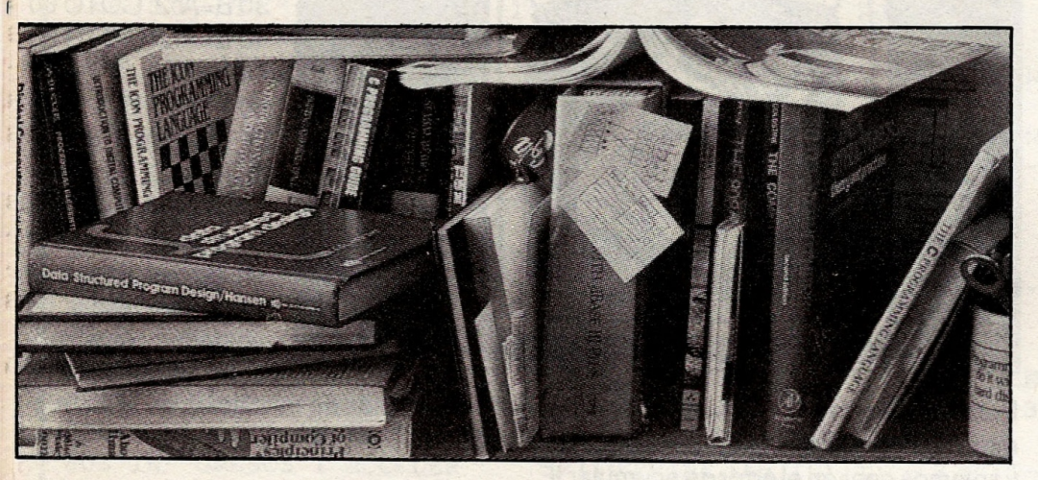
Variables

V: valor numérico
U: unidad del valor numérico
P: unidad a la que queremos pasar
R: resultado estimado por el alumno

```
4 CLS: KEY OFF
5 PRINT "KM=1,HEC=2,DAM=3,M=4,DM
=5,CM=6,MM=7"
6 PRINT
10 INPUT "INGRESA:VALOR Nrico,UN
IDAD,UNIDAD A  PASAR";V,U,P
11 PRINT
15 INPUT "INGRESA:RESULTADO";R
20 ON U GOTO 30,40,50,60,70,80,9
0
30 ON P GOTO 100,110,120,130,140
,150,160
40 ON P GOTO 170,100,110,120,130
,140,150,
50 ON P GOTO 180,170,100,110,120
,130,140
60 ON P GOTO 190,180,170,100,110
,120,130
70 ON P GOTO 200,190,180,170,100
,110,120
80 ON P GOTO 210200,190,180,170,
100,110
90 ON P GOTO 220,210,200,190,180
,170,100
100 END
110 G=V*10:GOTO 230
120 G=V*100:GOTO 230
130 G=V*1000:GOTO 230
140 G=V*10000:GOTO 230
150 G=V*100000!:GOTO 230
160 G=V*1000000#:GOTO 230
170 G=V/10:GOTO 230
180 G=V/100:GOTO 230
190 G=V/1000:GOTO 230
200 G=V/10000:GOTO 230
210 G=V/100000!:GOTO 230
220 G=V/1000000#:GQTO 230
230 IF R=G THEN LOCATE 12,12:PRI
NT "FELICITACIONES" ELSE 240
232 J$=INKEY$
234 IF J$="" THEN 232
236 GOTO 4
240 CLS:LOCATE 6,1:PRINT "KM HEC
 DAM M DM CM MM"
250 ON U GOTO 260,270,280,290,30
0,310,320
260 C$="1--0---0---0-0--0--0":GO
TO 340
270 C$="0--1---0---0-0--0--0":GO
TO 340
280 C$="0--0---1---0-0--0--0":GO
TO 340
290 C$="0--0---0---1-0--0--0":GO
TO 340
300 C$="0--0---0---0-1--0--0":GO
TO 340
310 C$="0--0---0---0-0--1--0":GO
TO 340
320 C$="0--0---0---0-0--0--1":GO
TO 340
340 LOCATE 6,2:PRINT C$
350 ON P GOTO 360,370,380,390,40
0,410,420
360 F$="KM -D-------------------
":GOTO 430
370 F$="HEC----D-----------------
":GOTO 430
380 F$="DAM----------D-----------
":GOTO 430
390 F$="M  ---------------D------
":GOTO 430
400 F$="DM -----------------D----
":GOTO 430
410 F$="CM --------------------D--
":GOTO 430
420 F$="MM ----------------------
D":GOTO 430
430 LOCATE 3,3:PRINT F$
440 LOCATE 5,5:PRINT "TENES QUE
MULTIPLICAR POR"
450 LOCATE 5,7:PRINT "EL Nx QUE
TE DA LA TABLA "
460 LOCATE 5,9:PRINT "(ATENCION
CON LA COMA-D)"
470 J$=INKEY$
480 IF J$="" THEN 470
490 CLS
500 PRINT "KM=1,HEC=2,DAM=3,M=4,
DM=5,CM=6,MM=7"
510 PRINT
520 PRINT "VALOR Nrico",V
530 PRINT "UNIDAD      ",U
540 PRINT "UNIDAD A PASAR";P
570 GOTO 11
```

MANEJO DE ARCHIVOS DESDE MSX BASIC: Archivos secuenciales. (Primera parte)

Nos han llegado algunas consultas sobre el manejo de archivos desde MSX Disk BASIC que se podrían resumir en una sola: el manual no especifica con un ejemplo que englobe todo lo referente al manejo de archivo y no trae un curso sobre este tema.



En efecto, no trae un curso (como tampoco lo trae el MSX BASIC) pues la idea del manual es que sirva de referencia para el usuario ya experimentado en el manejo del BASIC o archivos. Con esta nota trataremos de explicar más en detalle el uso de archivos en el entorno MSX.

Los españoles traducen el vocablo "file" de una forma más acertada que la que usamos en nuestro medio: los llaman ficheros en lugar de archivos.

En efecto, se puede pensar que un archivo es un fichero que posee fichas, y que estas fichas tienen los datos que deseamos almacenar (y luego recuperar).

Supongamos que tenemos un fichero con datos de los clientes de una empresa, donde constan los siguientes items:

Nombre:....................
Dirección:..................
Código Postal:....
Localidad:................
Provincia:..................
Teléfono:........

Una ficha común tendría estos datos escritos (preimpresos) de manera tal que el encargado de mantener el fichero actualizado sólo tiene que completar los lugares en blanco.

Comencemos a definir algunos conceptos.

Cada uno de los ítems incluidos en cada ficha se denomina "campo". Por ejemplo, tenemos el campo "Nombre", el campo "Dirección", etcétera.

El nombre del campo es la identificación que el usuario le da a ese conjunto de caracteres que forman el dato en cuestión, mientras que el contenido es el dato propiamente dicho.

Por ejemplo, nuestro fichero tiene un campo que se llama "Nombre", y este nombre es fijo para todos los registros. Sin embargo, el contenido del campo "Nombre" varía en cada caso, pudiendo ser "Juan Perez" o "BaireSoft S.A.".

Un conjunto de campos forman lo que se llama registro. En nuestro caso equivale a una ficha.

Finalmente, un conjunto de registros forman un archivo (o sea el fichero).

CAPITAL FEDERAL

Centro Cultural de la Ciudad de Buenos Aires
Taller Logo de computación
Junín 1930

Fundación de Informática y Educación
Centro de Computación Clínica
Asistencia al Usuario Discapacitado
Ramsay 2250 - Pabellón F
Tel. 784-2018

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692

Flores
Gral. Artigas 354 - Tel. 612-3902

Palermo
Guatemala 4733 - Tel. 71-4124

San Telmo
Chile 1345 - Tel. 37-0051 al 54

GRAN BUENOS AIRES

Castelar
C. Casares 997 - Tel. 629-2247

Lanús
Caaguazú 2186 - Tel. 247-0678

Morón
Belgrano 160 - Tel. 629-3347

Quilmes
Moreno 609 - Tel. 253-6086 al 89

Ramos Mejía
Bolívar 55 - 1er. piso - Tel. 658-4777

San Isidro
Av. Centenario 705 - Tel. 743-9678/747-6094

Vicente López
Av. Maipú 625 - Tel. 797-6720

Si hiciéramos un listado del contenido de un fichero completo, seguramente utilizaríamos las columnas para indicar el contenido de un campo y cada renglón sería un registro. Por ejemplo:

Listado de clientes

Nro.	Nombre	Dirección
001	Alberto Pérez	Balcarce 50
002	Juan García	Av de Mayo 1
003	Talent Computación	Chile 1347
...		

Como vemos, la analogía entre un fichero y un archivo de computadora es total. Hasta ahora hemos visto la manera en que se organiza la información dentro del archivo. Pero a esa información se debe acceder y esta debe ser grabada de alguna forma para que pueda ser procesada.

Existen dos métodos disponibles desde MSX Disk BASIC (con unidad de discos):

*- Archivos secuenciales.

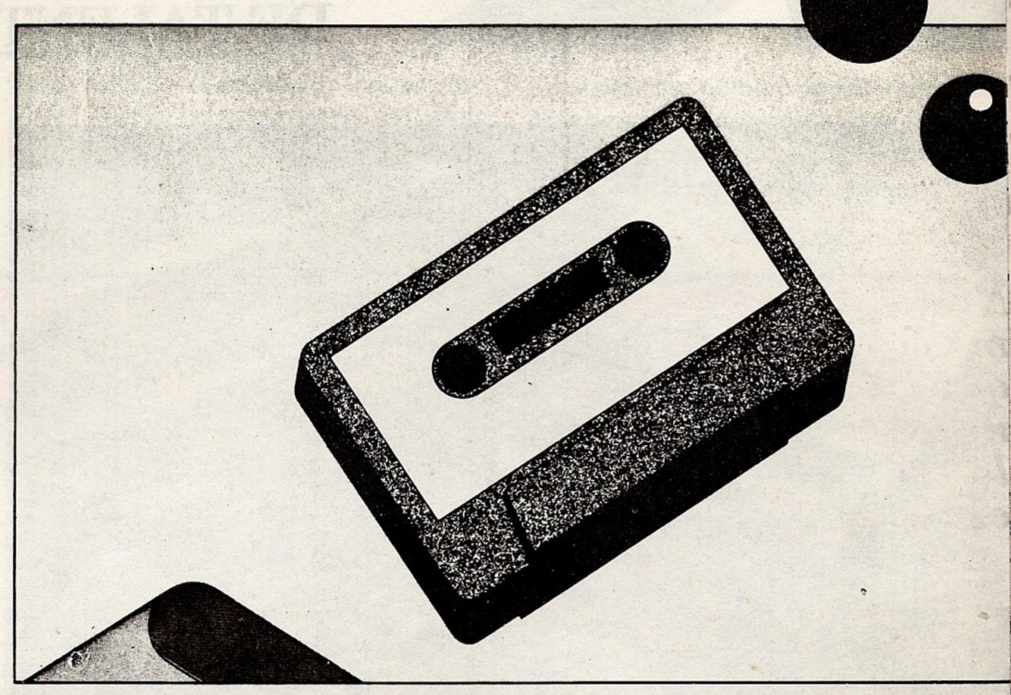
*- Archivos "random" o relativos.

En el caso de no disponer de una unidad de discos, los únicos archivos disponibles son los secuenciales. En este número nos dedicaremos a este tipo de archivos.

ARCHIVOS SECUENCIALES.

Estos archivos se llaman así puesto que los registros deben ser leídos en forma consecutiva, o sea primero leemos el registro 1, luego el registro 2, etcétera, hasta llegar al registro que deseamos obtener.

Los archivos que se pueden almacenar en cinta son de este formato por limitaciones propias de la cinta. El caso es totalmente análogo al de un casete de audio,



que debemos rebobinar hasta llegar al tema deseado, pasando previamente por todos los temas que lo preceden.

Para poder grabar y recuperar información almacenada en un casete o disquete, el MSX BASIC provee varias sentencias específicas.

Los pasos a seguir para poder grabar datos en un archivo secuencial son los siguientes:

1) Abrir el archivo.

Tal como haríamos con un fichero común, lo primero que tenemos que hacer es abrir el archivo (en el caso del fichero, abriríamos el cajón que lo contiene...).

La sentencia que utiliza el MSX BASIC es OPEN. Esta sentencia cumple varias tareas, a saber:

*- Inicializa el dispositivo para que luego reciba (o transmita) los datos.

*- Asigna al número indicado el archivo descripto por la sentencia. De esta forma, toda operación de entrada/salida puede referirse al "archivo Nro. 1" por ejemplo, y no al archivo "CAS:PEPE". Esto simplifica la operatoria del programa.

*- Define el modo de operación del archivo (INPUT, OUTPUT o APPEND).

En nuestro ejemplo crearemos un archivo que posea los números del 1 al 30.

El comando que incorporaremos al programa será:

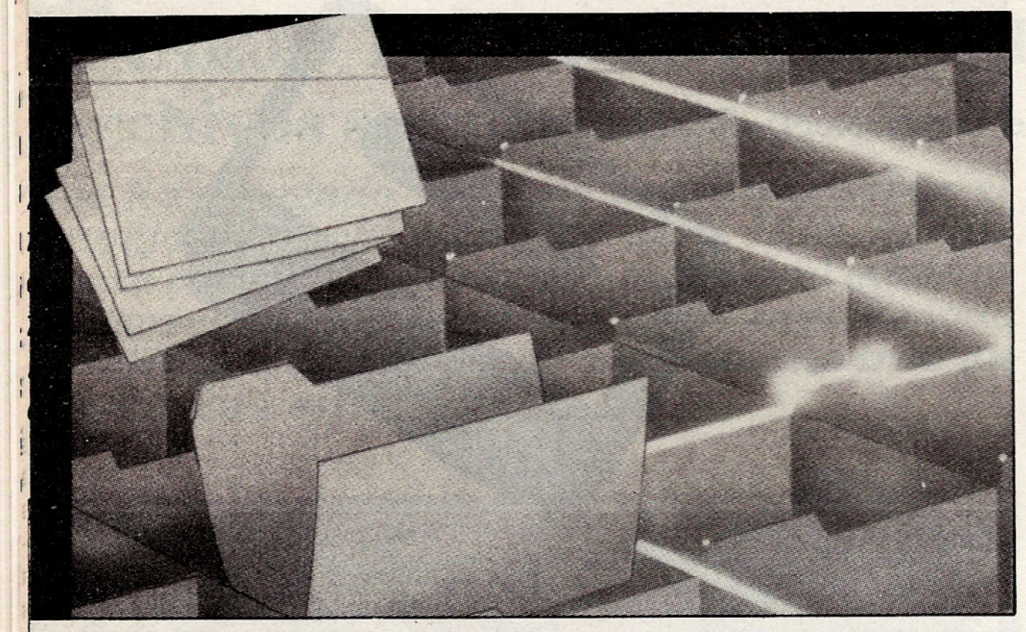
40 OPEN "PEPE" FOR OUTPUT AS#1

donde "PEPE" es el nombre del archivo, OUTPUT es la modalidad del archivo secuencial que graba los datos en el mismo (destruyendo todo dato que pudiese haber estado almacenado en el mismo) y #1 indica que el archivo Nro. 1 es en realidad el archivo PEPE.

2) Grabar los datos.

Para grabar datos en este archivo, debemos utilizar la sentencia PRINT# o PRINT# USING.

Esta sentencia se utiliza en forma totalmente análoga al clásico PRINT de pan-



alla, con la diferencia de que se debe indicar el Nro. de archivo a donde debe dirigirse la información.

Aquí debemos hacer una disquisición. Los archivos secuenciales permiten que se almacene cualquier tipo de dato dentro de diferentes registros. Es decir, podríamos (si queremos ser un poco desprolijos) almacenar en primer registro un número, luego un alfanumérico, posteriormente dos números, etcétera. Esto se debe a que el archivo secuencial permite que los registros sean de longitud variable.

Siguiendo nuestro ejemplo, para escribir los datos que deseamos sería:

```
50 FOR I=1 TO 30
60 'Aquí grabamos los datos creados
70 PRINT #1,I:PRINT I
80 NEXT I
```

3) Cerrar el archivo.

Finalmente, para que el archivo quede grabado y desconectado, deberemos utilizar la sentencia CLOSE. En nuestro ejemplo tenemos:

```
100 CLOSE #1:END
```

Bien. Este es nuestro primer programa, que permite la creación de un archivo PEPE con los números del 1 al 30:

Listado 1:

```
10 'Programa que graba PEPE
20 'Apertura del archivo secuencial
30 'en modo grabacion (OUTPUT).
40 OPEN "PEPE" FOR OUTPUT AS #1
50 FOR I=1 TO 30
60 'Aqui grabamos los datos creados
70 PRINT #1,I:PRINT I
80 NEXT I
90 'Close se encarga de cerrar y almacenar los datos pendientes.
100 CLOSE #1:END
```

Siguiendo con este ejemplo, podemos generar el programa que lee los datos que hemos creado en el programita anterior.

Nuevamente, deberemos seguir una secuencia preestablecida, a saber:

1) Abrir el archivo.

Igual que en el caso anterior, debemos abrir el archivo para poder usarlo. En este caso deseamos leer un archivo ya creado, que posee una cantidad x de datos (supondremos que no sabemos cuántos son, aunque en realidad son 30...).

Para ello, el modo a indicar en la sentencia OPEN es INPUT (entrada), de modo tal que la sentencia de apertura del archivo sería:

```
40 OPEN "PEPE" FOR INPUT AS #1
```

2) Leer los datos.

Para poder leer los datos, debemos verificar si realmente hay datos para leer, a fines de evitar que se genere el error "Input past end" (lectura después del fin de archivo), y una vez hecha esta verificación, recién utilizar la sentencia correspondiente a la lectura de datos.

Como ya habrá sospechado el lector perspicaz, si para escribir datos en un archivo usamos PRINT#, para leer datos del mismo utilizamos el comando INPUT#. De esta forma, para leer los datos que grabamos en el programa anterior, podríamos hacer lo siguiente:

```
90 INPUT #1,I:PRINT I
```

Vemos que el INPUT se realiza desde el archivo (#1) y que el PRINT es el común de pantalla, para visualizar el dato recién leído.

Pero anteriormente dijimos que debemos verificar si el archivo no ha llegado al final. Esta verificación la realiza la función EOF(<Nro. archivo>). El parámetro (lo que escribimos entre paréntesis) de esta función le indica a cuál de los archivos abiertos se refiere el programador. En nuestro caso, ingresaríamos la siguiente sentencia:

```
70 IF EOF(1) THEN 130
```

Y en la línea 130 haremos lo de costumbre cuando finalizamos las tareas con el archivo:

3) Cerrar el archivo.

```
130 CLOSE#1:END
```

Nuestro programa de captura de datos desde el archivo PEPE (creado en el ejemplo anterior) será:

Listado 2:

```
10 'Programa que lee PEPE
20 'Apertura del archivo secuencial
30 'en modo lectura (INPUT).
40 OPEN "PEPE" FOR INPUT AS#1
50 'Funcion EOF(1): controla si
60 'el archivo 1 finalizo
70 IF EOF(1) THEN 130
80 'Aqui leemos los datos grabados
90 INPUT#1,I:PRINT I
100 ' Volvemos a chequear el fin de archivo
110 GOTO 70
120 'Close se encarga de cerrar y almacenar los datos pendientes.
130 CLOSE#1:END
```

Finalmente existe otro modo de abrir un archivo de acceso secuencial: el modo APPEND o agregar.

Este modo permite agregar al final del archivo ya existente, más datos.

La metodología empleada en el programa para manejo del modo APPEND es idéntico al del modo OUTPUT, por lo que simplemente agregaremos el listado pertinente:

Listado 3:

```
10 'Programa que lee PEPE
20 'Apertura del archivo secuencial
30 'en modo grabacion al final (APPEND).
40 OPEN "PEPE" FOR APPEND AS#1
50 FOR I=50 TO 100
60 'Aqui grabamos los datos agregados
70 PRINT#1,I:PRINT I
```

80 NEXT I
90 'Close se encarga de cerrar y almacenar los datos pendientes.
100 CLOSE #1:END

En este caso, a continuación del número 30, seguirá el número 50 dentro del archivo hasta el número 100.

Para verificar el contenido del archivo PEPE al finalizar la ejecución del último ejemplo, podemos utilizar el programa del Listado 2 (modo INPUT) para visualizar los datos ya existentes y los agregados.

En el próximo número concluiremos con esta breve introducción al manejo de archivos, pero más específicos al uso del Disk BASIC: los archivos "random" o relativos.

INFOTALENT EN ACAMATICA

Desde mediados del año pasado estamos brindando a los usuarios de Talent MSX un servicio de consulta telefónica a través del teléfono 38-6601 InfoTalent, también conocido como "Hot Line".

A partir del mes de abril ampliamos este servicio de consultas ofreciendo una alternativa a través del Banco de Datos del ACA, ACAmática.

Nuestra identidad dentro del mismo es: T10003. Los invitamos a comunicarse con nosotros mediante la utilización del



Correo Electrónico de ACAmática, ACAmail.

Las consultas técnicas y pedidos de información en general serán contestados dentro de las 48 horas de emitido el correspondiente ACAmail.

Estamos estudiando la posibilidad de prestar un servicio similar a través de Siglo XXI y de Century 21, con el agregado de boletines informativos y la facilidad de rescatar programas y archivos de datos.

CONTESTANDO A LA HOT-LINE:

dBASE TRAVIESA

Soy usuario del dBASE II y me encontré con el siguiente problema:

Resulta ser que yo ingresé al dBASE y me puse a cargar unos datos que actualizaban mi base de datos. Luego de ello, le pedí que me listara los registros que cumplían con una condición, y me dio DISK FULL. Incluso intenté salir con QUIT para cerrar el archivo, y también me dio DISK FULL.

Finalmente, apagué el equipo, y al volver a cargar la base, me comenzó a dar RECORD OUT OF RANGE (registro fuera de rango). ¿Cómo lo soluciono?

```
Set Filter          Nest                Display             Exit   02:38:48 pm

Field Name              STATE                No combination
Operator                Matches              Combine with .AND.
Constant/Expression     "NY"                 Combine with .OR.
Connect                 .OR                  Combine with .AND..NOT.
                                             Combine with .OR..NOT.
Line Number             1

Line  Field         Operator              Constant/Expression   Connect
 1  ( STATE         Matches               "NY"                  .OR.
 2  | STATE         Matches               "DE"                ) .AND.
 3  ( PROD_DESC     Matches               "LM Bass Lures"     | .AND.
 4  | ORDER_DATE    More than or equal    11/01/85            )
 5
 6
 7
```

Respuesta:

"Lo que sucedió es que el dBASE perdió algunos punteros y la base de datos se corrompió", o sea, no se puede seguir usando en ese estado.

Para solucionarlo existe una función "mágica": PACK. Este comando no solo borra físicamente los registros marcados para tal fin, sino que además vuelve a calcular todos los punteros de la base solucionando el problema. Por lo tanto, los comandos a usar son:

USE <la base de datos en cuestión>
PACK

Luego de ello, la base volvió a la normalidad. Como acotación final queda una recomendación: una vez que se actualiza la base de datos, se debe grabar el archivo antes de utilizar cualquier otro comando. Si no queremos hacer QUIT, hagamos la siguiente secuencia:

Estando en modo edición, se pulsó <CTRL><W>.

Volviendo al modo comando, escribir:
USE

Este comando cierra el archivo, actualizándolo. Si queremos continuar operando, volvamos a escribir:
.USE <su base de datos>

LADRONES

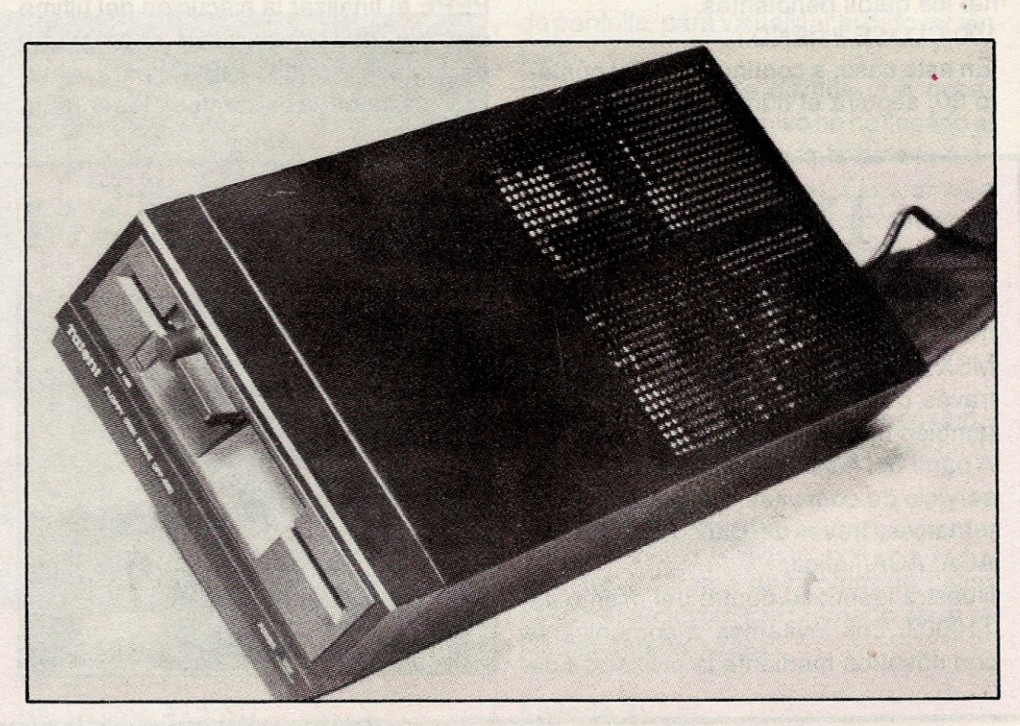
Soy usuario de la Talent TPC-310 y el BASIC tiene 28815 como siempre... ¿Dónde están los 128 kbytes?

Respuesta:

Están en la máquina a la espera de datos. Para utilizarlos, se debe recurrir al código de máquina o al RAM-Disk incorporado, ya que el BASIC no puede manejarlos. Esto se debe a que si el BASIC permitiera paginación, la velocidad de ejecución se vería notablemente disminuída... Sin embargo, podemos utilizarlo muy fácilmente. A continuación incorporamos un listadito de un programa que permite "POKEar" y "PEEKear" valores almacenados en la RAM dejada libre por el BASIC (los 32 kbytes de abajo). Para usar los otros 64 kbytes, más adelante daremos indicaciones al respecto.

Listado:

```
100 'Acceso a memoria en slots
110 '
120 SCREEN 0:WIDTH 38:KEY OFF
130 DEFUSR0=&HDFFF
150 FOR I=0 TO 55:READ D$
160 POKE &HE000+ I,VAL ("&H"+D$):
NEXT
170 '
180 PRINT
190 INPUT "SLOT NO. (0-3) ";S%
200 POKE &HE036,S%
210 INPUT "Escribe (0) o Lee(1) ";C%
220 POKE &HE035,C%
230 INPUT "Direccion (0000-8FFFH)
";A%
240 A%=VAL("&H"+A$)
250 IF A%<0 AND A%>-16385 THEN
230
260 IF C%>0 THEN 300
270 INPUT "Ingrese valor (00-FFH) ";D$
280 D%=VAL("&H"+D$):POKE
&HE037, D%
290 '
300 D=USR(A%)
310 '
320 IF C%=0 THEN PRINT "OK":GOTO
180
330 PRINT: D$= RIGHT$ ("0"+ HEX$ (D)
,2)
340 PRINT "Valor= ";D$:GOTO 180
350 '
360 DATA E5,D5,C5,F5,FE,02,20,28
370 DATA 23,23,5E,23,56,E5,EB,3A
380 DATA 36,E0,E6,03,F5,3A,35,E0
390 DATA B7,28,06,F1,CD,0C,00,18
400 DATA 09,3A,37,E0,5F,F1,CD,14
410 DATA 00,AF,E1,36,00,2B,77,FB
420 DATA F1,C1,D1,E1,C9,FF,00,00
```



Si algún memorioso duda si este programa ya lo vió, en efecto, hace un par de años salió publicado en K64. Volvemos a publicarlo aquí ya que los nuevos usuarios no se dieron por enterados de este programa (hace dos años no tenían la máquina...).

DESMEMORIADA

¿Existe alguna forma para que la unidad de discos me libere memoria?

Respuesta:

Como bien dice en el manual de uso de la unidad de discos (en castellano), para liberar memoria en el BASIC se debe pulsar la tecla <CTRL> cuando se enciende el sistema. Con esto se cancela el simulador de la segunda unidad, liberando 1,5 kbytes aproximadamente. Ya que estamos, podemos decir que si se pulsa la tecla <SHIFT> en las mismas condiciones, se cancela la unidad de discos.

HACIENDO TRAMPAS

¿Existe alguna manera de "engañar" a la máquina y hacer CALL SYSTEM aunque no haya inicializado el sistema con un disco con el MSX-DOS?

Respuesta:

Sí. Hay que ingresar la siguiente sentencia en el modo comando:
POKE &HF346,255
Luego de esto, podremos hacer CALL SYSTEM sin problemas.
CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

APRENDE SOLO

CLASE: EDUCATIVO

La capacidad de autoaprendizaje es una de las virtudes del siguiente programa, capaz de hacer deducciones a partir de datos que se le proporcionan.

Mucho se ha hablado sobre la capacidad de aprendizaje de las computadoras, e incluso se las hace aparecer en las películas como sistemas omnipotentes y con ansias de dominación.

La realidad es sustancialmente diferente. Un sistema capaz de pensar y/o tomar decisiones es de muy difícil realización.

El presente ejemplo es una muestra muy sencilla de este tipo de programas.

Para una utilización del mismo debemos teclear en respuesta a los requerimientos de entrada de datos lo siguiente:

NOMBRES: 5
CUALIDADES:3
NOMBRES (del 1 al 5):MICKEY, DONALD, TWEETY, SILVESTRE, PINK PANTHER
CUALIDADES (del 1 al 3):DISNEY, ALAS, ROSA

El programa luego hará una serie de preguntas y dará sus respuestas. En caso que de esté equivocado le tendremos que dar la correcta. Esta información se incorpora a sus datos para que no se repita el error.

Creemos que este puede ser un buen ejemplo, y de fácil comprensión, para animarse a entrar en el difícil mundo de la inteligencia artificial.



```
100 CLS
110 INPUT "CUANTOS NOMBRES
";NOMBRES
120 INPUT "CUANTAS
CUALIDADES";UAL
130 X=NOMBRES+UAL
140 DIM A$(NOMBRES):DIM
F$(UAL):DIM B(X)
150 FOR J=1 TO NOMBRES
160 J$=STR$(J)
170 PRINT "NOMBRE ";J$;
180 INPUT A$(J)
190 NEXT J
200 CLS
210 FOR J=1 TO UAL
220 J$=STR$(J)
230 PRINT "CUALIDAD ";J$;
240 INPUT F$(J)
250 NEXT J
1000 CLS
1010 PRINT "POSIBLES ELE
MENTOS"
1020 FOR J=1 TO NOMBRES
1030 PRINT A$(J)
1040 NEXT J
1050 PRINT "RESPONDA S/N"
1060 CUENTA=0
1070 X=.5
1080 FOR J=1 TO UAL
1090 X=X+X
1100 PRINT "SU CUALIDAD ES
";F$(J);" ?";
1110 INPUT Z$
1120 IF Z$="S" OR Z$="s" THEN
CUENTA=CUENTA+X
1130 NEXT J
1140 X=0
1150 FOR J=1 TO NOMBRES
1160 IF CUENTA=B(J) THEN X=J
1170 NEXT J
1180 IF X<>0 THEN 1260
1190 X=INT(RND(-
TIME)*NOMBRES)+1
1200 FLAG=0
1210 FOR J=1 TO NOMBRES
1220 IF B(J)=0 THEN 1240
1230 IF X=J AND
CUENTA<>B(J) THEN FLAG=1
1240 NEXT J
1250 IF FLAG=1 THEN 1190
1260 PRINT "ESTABAS PEN
SANDO EN ";A$(X);" ?";
1270 INPUT Z$
1280 IF Z$="S" OR Z$="s"
THEN B(X)=CUENTA:GOTO 1000
1290 INPUT "CUAL ERA LA
RESPUESTA CORRECTA ";Z$
1300 FOR J=1 TO NOMBRES
1310 IF A$(J)=Z$ THEN
B(J)=CUENTA
1320 NEXT J
1330 GOTO 1000
```

LA AVENTURA DEL PENSAMIENTO

El desarrollo de software requiere de un proceso de varias etapas. A través de un ejemplo, un programa que hace gráficos de barras, ofrecemos las experiencias de un grupo de jóvenes sobre el tema.

En varios artículos periodísticos se ha señalado el riesgo que conlleva el uso inadecuado de las nuevas técnicas derivadas de la microelectrónica, dentro de las cuales puede incluirse la computación personal.

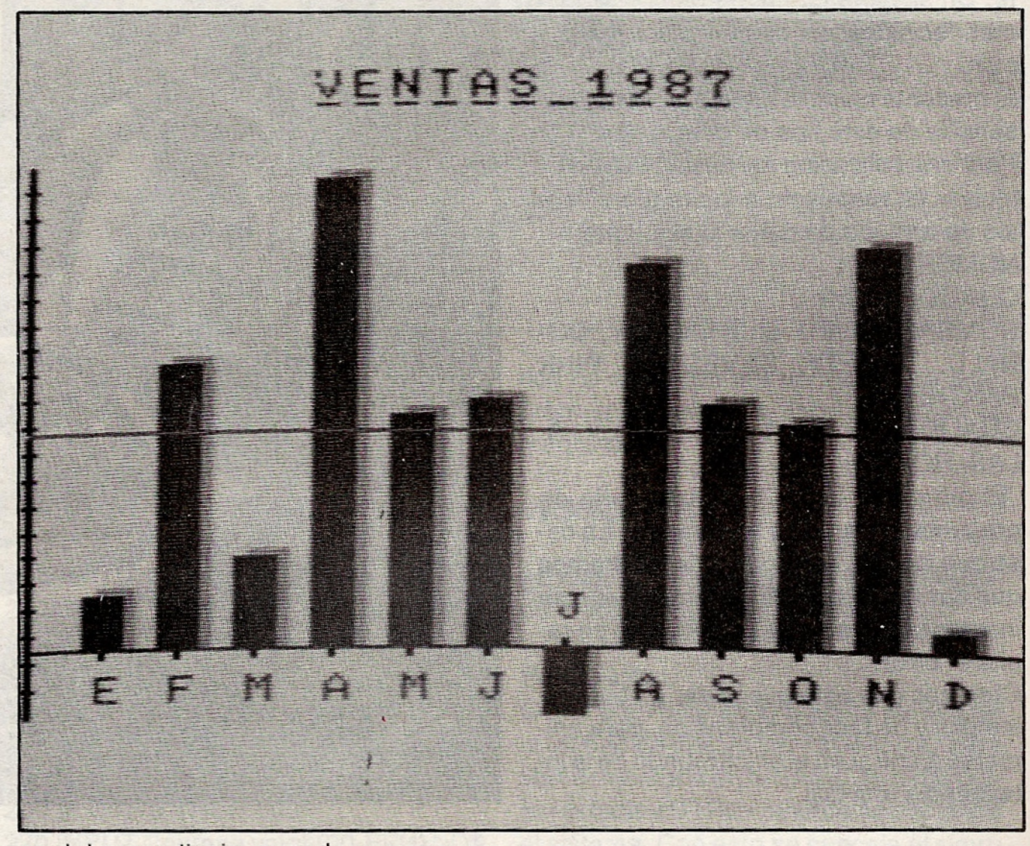
EL PENSAMIENTO REFLEXIVO

La fascinación que otorga la rapidez y exactitud con que se efectúan procesos numéricos y no-numéricos, y la sensación de poder que surge del uso de estos modernos artefactos electrónicos, puede llevar en sus mismas entrañas un velo que oculte la aptitud más importante del ser humano: el pensamiento reflexivo.

El temor que ha sido puesto de manifiesto, por los autores de los artículos que mencionábamos al principio, está dado por suponer que el uso de computadoras lleve a eliminar lo poco que aún resta del hábito de pensar. Además esto ha sido alimentado por la asociación que se hace inconcientemente: computadoras - videojuegos, y por el hecho de que los primeros pasos se dieron en gran medida, de la mano de quienes provenían del área de Sistemas (analistas, programadores, etcetera) y no de verdaderos docentes.

Quienes nos dedicamos a la difícil tarea de iniciar a jóvenes y adultos en el uso de las microcomputadoras hogareñas, no podemos perder de vista la importancia que la meditación y la reflexión poseen para las mentes de los "aprendices", cualquiera sea su edad, ni tampoco el hecho de que "la meditación rehuye la prisa".

Es por ese motivo que personalmente trato de no apurar el desarrollo de una clase en aras del formal cumplimiento de un cronograma preelaborado. Tampoco de pasar por alto las inquietudes que jóvenes y adultos plantean en el transcurso del aprendizaje, cuando una vez que se han ambientado en mi modesto taller, se animan a cuestionar sobre la utilidad o la importancia de cada paso que se da, sobre la estructura de algún procedimiento, o aún sobre consideraciones estéticas de los programas.



EL GRAFICO DE BARRAS

En el transcurso de las últimas clases del ciclo de un grupo de BASIC Jóvenes, al efecto de aplicar el tema de manejo de VECTORES a través de un ejemplo significativo, comenzamos a desarrollar un programa cuya finalidad consistía en trazar en SCREEN2 un GRAFICO DE BARRAS.

El ejercicio en sí no se destacaba por su originalidad, pero la sorpresa (mía) fue el interés que despertó en el grupo, a medida que íbamos avanzando, y en las propuestas que surgían a cada paso que dábamos.

Cada pregunta o propuesta de modificación que hacían los alumnos era tratada con detenimiento, analizada, MEDITADA, reflexionada en conjunto sobre la conveniencia o no de su incorporación, y en caso de su aceptación, se analizaba con sumo cuidado COMO efectuarla y DONDE insertarla.

Cada paso que se daba en este sentido sentaba las bases de autoconfianza (tal vez por la valoración que yo como docente hacía de cada propuesta o sugerencia, no importaba lo inusual que pareciera) en los jóvenes, quienes iban adquiriendo soltura y aplomo para nuevas instancias.

Del proyecto original se llegó a otro bastante diferente en cuanto al nivel de calidad y profundidad del mismo.

A este desarrollo lo llamamos con cariño "LA AVENTURA DEL PENSAMIENTO".

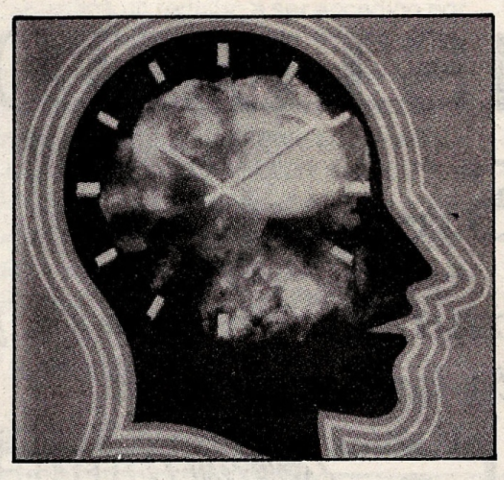
porque les permitió comprender que con imaginación y detenida reflexión, sin apuros, se puede lograr alcanzar un excelente nivel cualitativo en programación y, fundamentalmente, podemos cultivar ese don tan preciado que constituye el pensamiento.

EL PROGRAMA EN SI

El programa cuenta con seis módulos con diversos grados de dificultad y originalidad, que en esencia permiten ingresar datos de los doce meses de un año que son graficados a escala con barras verticales proporcionadas.

Los aspectos más destacables del mismo, teniendo en cuenta el nivel de los alumnos, son:

a) Al ingresar el TITULO del gráfico, una pequeña rutina transforma en MAYUSCULAS cualquier letra que se ingrese en minúsculas.



b) Admite datos numéricos NEGATIVOS, que son representados con barras invertidas.

c) Las barras son PROPORCIONALES al valor del dato correspondiente (se trabajó a escala).

d) El eje horizontal es DESPLAZABLE sobre el vertical, proporcionalmente a la magnitud de valores negativos.

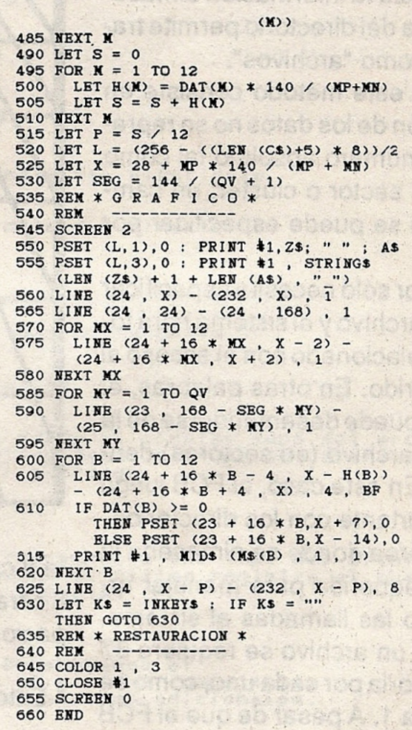
e) Las iniciales de cada uno de los meses se colocan por debajo o por encima del eje horizontal, de acuerdo al signo positivo o negativo del valor correspondiente, para no interferir con las barras.

f) El manejo de tres modos de pantalla diferente: SCREEN 3 para el título, SCREEN 0 para el ingreso de datos y SCREEN 2 para el gráfico en sí.

g) La validación sistemática que se hizo en los ingresos de datos.

h) El desarrollo estructurado del programa y el uso adecuado de REMarks, indentado de ciclos FOR/NEXT y de IF/THEN, etcétera, que facilitó notablemente la depuración que se hizo del mismo.

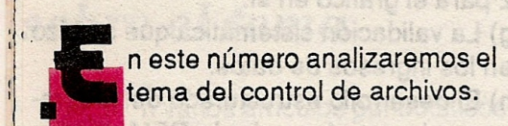
Gustavo Delfino



```
100 REM  **********************
105 REM  *                    *
110 REM  * GRAFICO DE BARRAS *
115 REM  *                    *
120 REM  **********************
125 REM
130 REM  por Silvia RODRIGUEZ
135 REM     Veronica JAIT
140 REM     Fernando BASILE
145 REM
150 REM  Prof. Gustavo Delfino
155 REM
160 REM * ASIGNACIONES *
165 REM   ------------
170 DIM L$(30) , M$(12)
175 DIM DAT(12) , H(12)
180 FOR M = 1 TO 12
185   READ M$(M)
190 NEXT M
195 DATA ENERO, FEBRERO, MARZO, ABRIL, MAYO,
    JUNIO, JULIO, AGOSTO, SETIEMBRE, OCTUBRE
    , NOVIEMBRE, DICIEMBRE
210 REM * C A R T E L *
215 REM   -----------
220 OPEN "GRP:" AS #1
225 COLOR 8 , 11 , 11
230 SCREEN 3
235 PSET (20,30),0 : PRINT #1,"GRAFICO".
240 PSET (96,90),0 : PRINT #1,"DE"
245 PSET (30,150),0 : PRINT #1,"BARRAS"
250 IF INKEY$ = "" THEN GOTO 250
255 REM * ACONDICIONAMIENTO *
260 REM   -----------------
265 SCREEN 0
270 WIDTH 40
275 KEY OFF
280 COLOR 4 , 15 , 15
285 REM * INGRESO DE DATOS *
290 REM   ----------------
295 LOCATE 9,1:PRINT "GRAFICO DE BARRAS"
300 LOCATE 0,4 : PRINT "Título ?"
305 LOCATE 0,6 : INPUT C$
310 IF C$ = "" THEN GOTO 305
315 FOR L = 1 TO LEN (C$)
320   LET L$(L) = MID$ (C$ , L , 1)
325   LET COD = ASC (L$(L))
330   IF COD < 123 AND COD > 96
         THEN LET COD = COD - 32
335   LET Z$ = Z$ + CHR$ (COD)
340 NEXT L
345 LOCATE 0 , 9 : PRINT "Qué año ?"
350 LOCATE 0 , 11 : INPUT A$
355 IF LEN(A$) > 4 OR LEN (A$) < 2
       THEN BEEP : GOTO 350
360 IF VAL (A$) = 0 THEN GOTO 350
365 LET A$ = "19"+MID$(A$,LEN(A$) - 1,2)
370 LOCATE 0,14 : PRINT "Marcas eje Y ?
375 LOCATE 0,16 : INPUT QV
380 IF QV < 0 THEN LET QV = 0
385 CLS
390 LOCATE 1,1 : PRINT "Ingresa los
    DATOS correspondientes a:"
395 FOR MM = 1 TO 12
400   LOCATE 2,MM + 2 : PRINT MM ;
      TAB (6) ; M$(MM) ;
405   LOCATE 17 : INPUT DAT(MM)
410 NEXT MM
415 LOCATE 2 , 16 : PRINT "Algún error
    en los datos ? (S/N)" ;
420 LOCATE 28 : LET R$ = INPUT$(1)
425 IF NOT (R$ = "S" OR R$ = "s")
       THEN GOTO 455
430 LOCATE 2,18 : PRINT "En qué mes ?
    (1 a 12)" ;
435 LOCATE 25 : INPUT E
440 IF E < 1 OR E > 12 THEN GOTO 435
445 LOCATE 17 , E + 2 : INPUT DAT(E)
450 GOTO 415
455 REM * C A L C U L O S *
460 REM   -----------------
465 LET MP = 0
470 LET MN = 0
475 FOR M = 1 TO 12
480   IF DAT(M) >= 0
        THEN IF DAT(M) > MP
               THEN LET MP = DAT(M)
               ELSE GOTO 485
        ELSE IF ABS (DAT(M)) > MN
               THEN LET MN = ABS (DAT
                                   (M))
485 NEXT M
490 LET S = 0
495 FOR M = 1 TO 12
500   LET H(M) = DAT(M) * 140 / (MP+MN)
505   LET S = S + H(M)
510 NEXT M
515 LET P = S / 12
520 LET L = (256 - ((LEN (C$)+5) * 8))/2
525 LET X = 28 + MP * 140 / (MP + MN)
530 LET SEG = 144 / (QV + 1)
535 REM * G R A F I C O *
540 REM   -------------
545 SCREEN 2
550 PSET (L,1),0 : PRINT #1,Z$; " " ; A$
555 PSET (L,3),0 : PRINT #1 , STRING$
    (LEN (Z$) + 1 + LEN (A$) , "_")
560 LINE (24 , X) - (232 , X) , 1
565 LINE (24 , 24) - (24 , 168) , 1
570 FOR MX = 1 TO 12
575   LINE (24 + 16 * MX , X - 2) -
      (24 + 16 * MX , X + 2) , 1
580 NEXT MX
585 FOR MY = 1 TO QV
590   LINE (23 , 168 - SEG * MY) -
      (25 , 168 - SEG * MY) , 1
595 NEXT MY
600 FOR B = 1 TO 12
605   LINE (24 + 16 * B - 4 , X - H(B))
      - (24 + 16 * B + 4 , X) , 4 , BF
610   IF DAT(B) >= 0
        THEN PSET (23 + 16 * B,X + 7),0
        ELSE PSET (23 + 16 * B,X - 14),0
615   PRINT #1 , MID$ (M$(B) , 1 , 1)
620 NEXT B
625 LINE (24 , X - P) - (232 , X - P), 8
630 LET K$ = INKEY$ : IF K$ = ""
    THEN GOTO 630
635 REM * RESTAURACION *
640 REM   -------------
645 COLOR 1 , 3
650 CLOSE #1
655 SCREEN 0
660 END
```

LOS MISTERIOS DEL MSX-DOS (3ra parte)

Seguimos investigando el MSX-DOS, para poner luz sobre esta importante herramienta.



En este número analizaremos el tema del control de archivos.

ACCESO DE ARCHIVO

* FCB (File Control Block - Bloque de Control de Archivo).

Cuando se utiliza la información almacenada en el área del directorio permite tratar los datos como "archivos".
La ventaja de este método consiste en que la ubicación de los datos no se representa con un número absoluto tal como un número de sector o cluster; en cambio, el archivo se puede especificar por su "nombre".
El programador sólo necesita especificar el nombre de archivo y el sistema hará todo el trabajo relacionado con el acceso al archivo requerido. En otras palabras, el programador puede desentenderse de la ubicación del archivo (en sectores) dentro del disco. En este caso, el FCB juega un papel importante con los directorios.
El FCB es el área donde se almacena la información requerida para manejar archivos usando las llamadas al sistema. Para manejar un archivo se requiere 37 bytes de memoria por cada uno, como se ve en la Figura 1. A pesar de que el FCB se puede colocar en cualquier posición de memoria, la dirección 005Ch es la que utiliza normalmente el MSX-DOS.

Veamos la descripción:

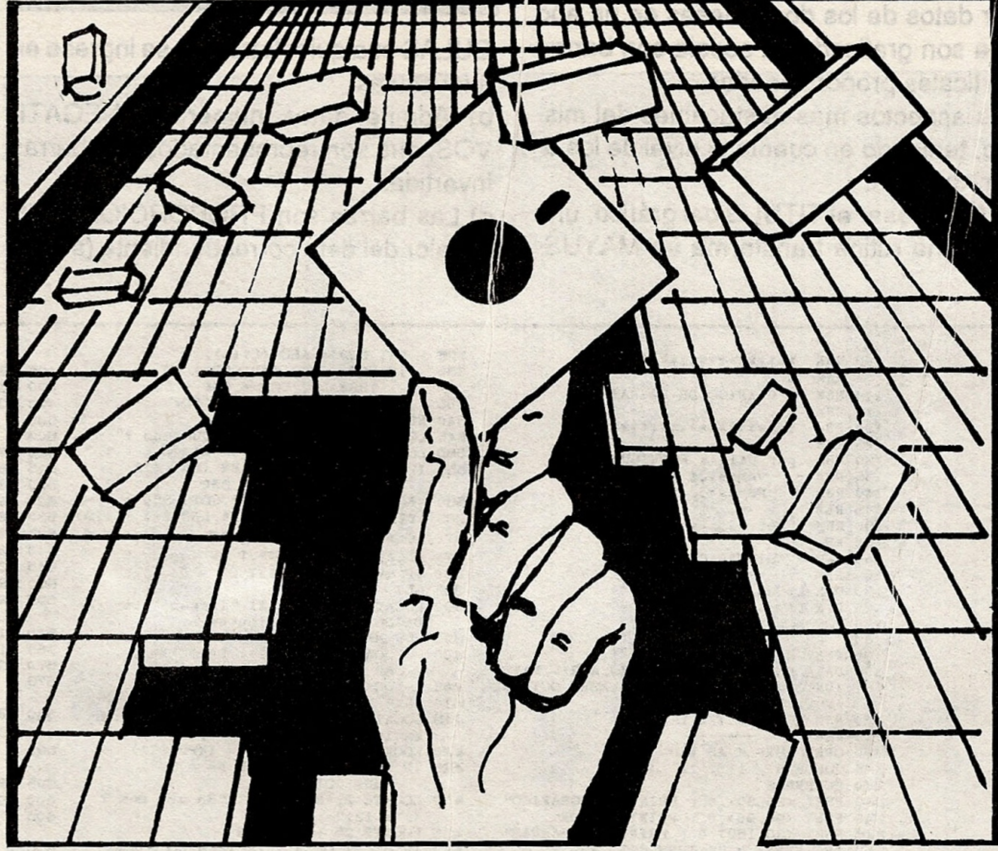
* Número de drive (00h)

Indica la unidad de discos que contiene el archivo.
Se describe así:

0 = Unidad activa
1 = Unidad A:
2 = Unidad B:

* Nombre de archivo (01h a 08h)

Un nombre de archivo puede tener hasta 8 caracteres. Cuando tiene menos de 8 caracteres, el resto se completa con espacios (caracter 20h).



* Extensión (09h a 0Bh)

La extensión puede tener hasta 3 caracteres. Cuando tiene menos de 3, el resto se completa con espacios.

* Bloque en uso (0Ch a 0Dh)

Indica el número de bloque que está siendo referido actualmente por un acceso secuencial (ver la llamada al sistema 14h y 15h).

* Tamaño del registro (0Eh a 0Fh)

Especifica el tamaño de la unidad de datos (registro) que será leído o grabado cuando se acceda, en bytes (ver las llamadas al sistema 14h, 15h, 21h, 27h, 28h)

* Tamaño del archivo (10h a 13h)

Indica el tamaño del archivo en bytes.

* Fecha (14h a 15h)

Indica la fecha de la última vez que se grabó el archivo. El formato es el mismo utilizado en el directorio.

* Hora (16h a 17h)

Indica la hora de la última vez que se grabó el archivo. El formato es el mismo utilizado en el directorio.

* ID del dispositivo (18h)

Cuando se abre un periférico como archivo, el valor indicado en la tabla 1 se espe-

cifica en este campo ID del dispositivo. Para los archivos normales de disco, el valor de este campo es 40h + número de drive. Por ejemplo, el ID del dispositivo para el drive A es 40h (para futuras expansiones, los programas de aplicación no deben utilizar este byte de ID).

Tabla 1 - ID de dispositivo

Nombre dispositivo	ID
CON (Consola)	0FFh
PRN (Impresora)	0FBh
LST (List=Impresora)	0FCh
AUX (Auxiliar)	0FEh
NUL (Nulo)	0FDh

* Ubicación del directorio (19h)

Indica el orden de las entradas de directorio de un archivo en el área de directorio.

* Cluster tope (1Ah a 1Bh)

Indica el cluster tope del archivo en el disco.

* Ultimo cluster accedido (1Ch a 1Dh)

Indica el último cluster del archivo al que se accedió.

* Ubicación relativa desde el cluster tope del último cluster accedido (1Eh a 1Fh)

Indica la ubicación relativa desde el cluster tope hasta el último cluster accedido.

* Registro actual (20h)

Indica el registro que está siendo accedido actualmente referido al acceso secuencial (ver llamadas al sistema 14h y 15h)

* Registro random (21h a 24h)

Especifica un registro a ser accedido por el acceso random o por el acceso a bloque random. Especificando un valor de 1 a 63 para el campo de tamaño de registro descripto anteriormente provoca la utilización de los cuatro bytes desde 21h a 24h, teniendo los bytes desde 21h a 23h significado sólo cuando el tamaño del registro es mayor que 63 (ver llamados al sistema 14h, 15h, 21h, 22h, 27h, 28h)

* Abriendo un archivo

Se requiere un procedimiento especial para abrir un archivo usando FCB. "Abriendo un archivo" significa, a nivel de llamadas del sistema, transformando un FCB incompleto cuyo campo de nombre de archivo es lo único definido a un FCB completo, usando la información escrita en el directorio.
La Figura 2 muestra la diferencia entre un FCB sin abrir y otro abierto

* Cerrando un archivo.

Figura 1 - Organización del FCB

```
FCB
Bytes desde el
tope
*---*
| 0| Número de drive
*---*
| 1| Nombre de archivo
| .|     nombre .... 8 bytes
| .|     extensión.. 3 bytes
|11|
*---*
|12| Bloque en uso
|13|   cantidad de bloques desde el tope del
*---*  archivo hasta el bloque en uso
|14| Tamaño del registro
|15|   1 a 65535
*---*
|16| Tamaño del archivo
| .|   1 a 4294967296
| .|
| .|
|19|
*---*
|20| Fecha
|21|   Con el mismo formato del directorio
*---*
|22| Hora
|23|   Con el mismo formato del directorio
*---*
|24| ID de dispositivo
*---*
|25| Ubicación del directorio
*---*
|26| Número de cluster tope del archivo
|27|
*---*
|28| Ultimo cluster accedido
|29|
*---*
|30| Ubicación relativa desde el cluster tope del archivo
|31|   Cantidad de clusters desde el tope del archivo hasta
*---*  el último cluster accedido
|32| Registro actual
*---*
|33|
| .| Registro random
| .|   Orden del registro respecto del tope del archivo
| .|   Normalmente almacena el último registro al que se
|36|   accedió en forma directa.
*---*

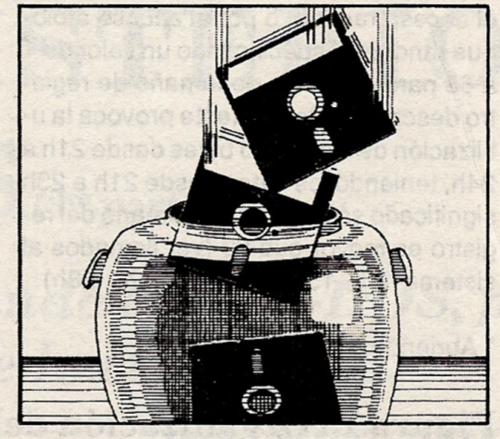
Nota:
       El uso del FCB difiere, dependiendo si
       se utilizan las llamadas al sistema compatibles
       con CP/M o las llamadas al sistema adicionales.
```

Cuando un archivo se abre y escribe, el contenido de cada campo del FCB, como por ejemplo, el tamaño de archivo, también se modifica. A menos que la información del FCB sin actualizar se devuelva al área del directorio, la información del directorio y el contenido actual del archivo puede ser diferente en el siguiente acceso.

Justamente, la operación de devolver el FCB actualizado al directorio corresponde al cierre de un archivo a nivel de llamados de sistema.

* Acceso directo por bloques (Manejo de archivos con registros).

MSX-DOS posee dos llamadas al sistema con acceso directo: "RANDOM BLOCK READ" y "RANDOM BLOCK WRITE" (Lectura directa de bloques y Escritura directa de bloques). Con estas llamadas al sistema, un archivo se puede dividir en unidades de datos de cualquier tamaño, que pueden manejarse mediante números, como 0, 1, 2,..., desde el principio.
Esta unidad de datos se denomina "registro". El tamaño del registro puede ser cualquier valor mayor a un byte. Por lo



tanto, se puede tratar a todo un archivo como un registro (acceso secuencial extremo), tratar datos con un byte como un registro (acceso directo extremo), o tratar 128 bytes como un registro (estilo CP/M), son todos posibles.
En este caso, los campos FCB, "tamaño de registro" y "registro random" se utilizan para especificar un registro. El valor del campo "tamaño del registro" es la cantidad de bytes en un registro. Los campos de registro random pueden tener cualquier cantidad de registros para ser accedidos.

* Acceso secuencial (Manejo de archivos por registro de longitud fija + registro actual + bloque actual).

MSX-DOS puede también acceder de la misma forma que CP/M para poder mantener compatibilidad. Una forma es el archivo secuencial que se maneja con "registro actual" y "bloque actual". Esto utiliza registros de longitud fija de 128 bytes como la unidad básica de datos. El acceso al archivo se hace siempre desde el tope secuencialmente y el número de registros que fue accedido se cuenta en el campo del registro actual del FCB. El valor del registro actual se coloca en 0 cuando llega a 128, y se lleva el acarreo al campo de bloque actual.

* Acceso random (manejo de archivo con registros de longitud fija + registro random)

Un segundo método incluido para mantener compatibilidad con CP/M es un método de acceso directo usando campos de registro random. Se puede acceder a un registro en cualquier ubicación, pero el tamaño del registro es fijo, de 128 bytes.
En el próximo número finalizaremos con la investigación del MSX-DOS, e incluiremos la lista de los llamados al sistema y un programa de ejemplo.

Por Hugo Daniel Caro

Figura 2 - Antes/después de abrir un FCB

```
Antes de abrir

    *-*--------
 0  |S| Número de drive
    *-*--------
 1  |S|   ^
    *-*   |
 2  |S|   |
    *-*   |
 3  |S|   |
    *-* Nombre
 4  |S|  del
    *-* archivo
 5  |S|   |
    *-*   |
 6  |S|   |
    *-*   |
 7  |S|   |
    *-*   |
 8  |S|   |
    *-*   |
 9  |S|   |
    *-*   |
10  |S|   |
    *-*   |
11  |S|   v
    *-*-------
12  | |
    *-*
13  | |
    *-*
14  | |
    *-*
15  | |
    *-*
16  | |
    *-*
17  | |
    *-*
18  | |
    *-*
19  | |
    *-*
20  | |
    *-*
21  | |
    *-*
22  | |
    *-*
23  | |
    *-*
24  | |
    *-*
25  | |
    *-*
26  | |
    *-*
27  | |
28  | |
    *-*
29  | |
    *-*
30  | |
    *-*
31  | |
    *-*
32  | |
    *-*
33  | |
    *-*
34  | |
    *-*
35  | |
    *-*
36  | |
    *-*

Después de abrir

    *-*--------
 0  |S| Número de drive se
        convierte el activo
        (00h) por el real
        (01h a 08h)
    *-*--------
 1  |S|   ^
    *-*   |
 2  |S|   |
    *-*   |
 3  |S|   |
    *-* Nombre
 4  |S|  del
    *-* archivo
 5  |S|   |
    *-*   |
 6  |S|   |
    *-*   |
 7  |S|   |
    *-*   |
 8  |S|   |
    *-*   |
 9  |S|   |
    *-*   |
10  |S|   |
    *-*   |
11  |S|   v
    *-*------
12  | |
    *-* Bloque actual
13  | |
    *-*------
14  | |
    *-* Tamaño del registro
15  | |
    *-*------
16  |S|  ^
    *-*  |
17  |S|
    *-* Tamaño del archivo
18  |S|
    *-*  |
19  |S|  v
    *-*------
20  |S|
    *-* Fecha
21  |S|
    *-*------
22  |S|
    *-* Hora
23  |S|
    *-*------
24  |S| ID del dispositivo
    *-*------
25  |S| Ubicación en directorio
    *-*------
26  |S|
    *-* Cluster tope del archivo
27  |S|
    *-*------
28  |S|
    *-* Ultimo número de cluster
         accedido
29  |S|
    *-*------
30  |S|
    *-* Ubicación relativa desde
        el cluster tope del archivo
31  |S|
    *-*------
32  | | Registro actual
    *-*------
33  | |  ^
    *-*  |
34  | |
    *-*  Registro random
35  | |
    *-*  |
36  | |  v
    *-*------

*-*
|S| Seteado
*-*

*-*
| | No seteado
*-*
```

PROGRAMA 1

```
COMMENT *

        COMANDO EXTERNO PARA CARGAR
        EL SISTEMA CP/M DESDE EL
        MSX-DOS PARA SVI-707.

        R.C.  Diciembre 1987

         *
;
        .Z80
;
               LD      HL,INIC          ; traslado del núcleo
               LD      DE,8000H         ; del programa a pá-
               LD      BC,FIN-INIC+1    ; gina 2, slot 1
               LDIR
               JP      8000H
;
        INIC:  LD      DE,MENS
               LD      C,9              ; rutina 'STRING OUT'
               CALL    5
               LD      C,8              ; rutina 'DIRECT INPUT'
        LOOP:  CALL    5
               CP      13
               JR      NZ,LOOP
               LD      A,5CH            ; página 1, slot 3
               OUT     (0A8H),A
               LD      A,0FFH
               LD      (7FBEH),A        ; 'I/O port drive'
               RST     0                ; 'warm boot'
;
        MENS:  DEFM    'Introducir un disco con el sistema CP/M',10,13
               DEFM    'y pulsar ''ENTER'' para cargarlo',10,13
               DEFM    'o ''CTRL-C'' para salir',10,13,'$'
        FIN:
               END
```

PROGRAMA 3

```
* COLOR.CMD
* Esta rutina es utilizable en dBASE II
STORE 'COLOR' TO COLOR
POKE 43100,243,62,1,33,233,243,119,35,62,7,119,35,119,253,42,192
POKE 43116,252,221,33,98,0,205,28,0,251,201
SET CALL TO 43100
CALL COLOR
RETURN
;COLOR.MAC
;Esta rutina es compilable. Genera un archivo COLOR.COM
;ejecutable en el MSXDOS.
 .Z80
                ASEG
        ORG     0100H
CHGCLR  EQU     0062H
CALSLT  EQU     001CH
EXPTBL  EQU     0FCC1H
;
        DI
        LD      A,1                 ;Color frente negro
        LD      HL,0F3E9H
        LD      (HL),A
        INC     HL
        LD      A,7                 ;Color fondo cyan
        LD      (HL),A
        INC     HL
        LD      (HL),A              ;Color borde cyan
        LD      IY,(EXPTBL-1)       ;Slot de la main ROM
        LD      IX,CHGCLR           ;Subrutina change color en ROM
        CALL    CALSLT
        EI
        RET
        END
```

PROGRAMA 2

```
        ORG     0D000H
;
        LD      HL,MENS
L1:     LD      A,(HL)
        CP      '$'
        JR      Z,L2
        RST     18H
        INC     HL
        JR      L1
L2:     CALL    9FH
        CP      3
        RET     Z
        CP      13
        JR      NZ,L2
        LD      A,5CH           ; página 1, slot 3
        OUT     (0A8H),A
        LD      A,0FFH
        LD      (7FBEH),A       ; 'I/O port drive'
        RST     0               ; 'warm boot'
MENS:   DEFM    'Introducir un disco con el sistema CP/M'
        DEFM    'y pulsar ''ENTER'' para cargarlo'
        DEFB    10,13
        DEFM    'o ''CTRL-C'' para salir$'
```

¿Cómo cargar el CP/M desde el MSX-DOS para la disquetera SVI-707?

La disquetera SVI-707 posee en su ROM un "booter" para la carga del sistema operativo CP/M 80, el cual solo trabaja al encender la máquina.

En la serie de comandos externos que se incluyen en el disco de este sistema hay uno (el GOMSX.COM) que permite cargar el MSX-DOS desde el CP/M.

Los programas 1 y 2 permiten "rebootear" el CP/M desde el MSX-DOS o desde el BASIC sin necesidad de apagar y encender la computadora.

El listado del programa 1 funciona bajo el MSX-DOS como comando externo (extensión .COM). Debe ser ensamblado con el Macro-80 y luego reubicado en la dirección hexadecimal 100 con el Link-80.

El segundo programa trabaja en la memoria alta de la máquina y puede ser cargado desde el BASIC mediante BLOAD"nombre",R

(Colaboración de Roberto C. Civitillo)

¿Cómo cambiar de color en DBASE II?

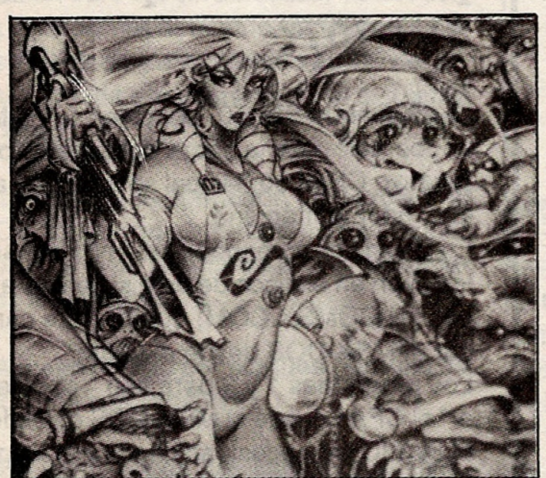
Esta rutina permite cambiar el color de frente y de fondo como parte de un programa en DBASE II.

Como se sabe, el DBASE II no tiene una instrucción que lo haga, por lo que esta pequeña rutina en código máquina puede insertarse como parte de cualquier programa o invocarla desde el DBASE II como DO COLOR.(Ver programa 3)

Más abajo puede verse el listado 3 desensamblado de esta rutina. La única diferencia entre ambas es que en DBASE II el programa corre a partir de la dirección 43100 (decimal) y el otro a partir de 0100 (256 decimal) y que es compilable para crear el archivo COLOR.COM

(Colaboración del Ing. Carlos Atashian)

PHANTIS



La Comandante Serena tiene por misión rescatar a su compañero de expedición capturado en la Luna 4 del sistema SOTPOK, más conocido como el mundo de PHANTIS, un paraje insólito habitado por una fauna hostil y una flora exhuberante.

Este programa tiene dos partes, cada una grabada en una de las caras de la cinta. Para jugar a la segunda, es necesario descubrir el código de acceso que se encuentra al terminar la primera.

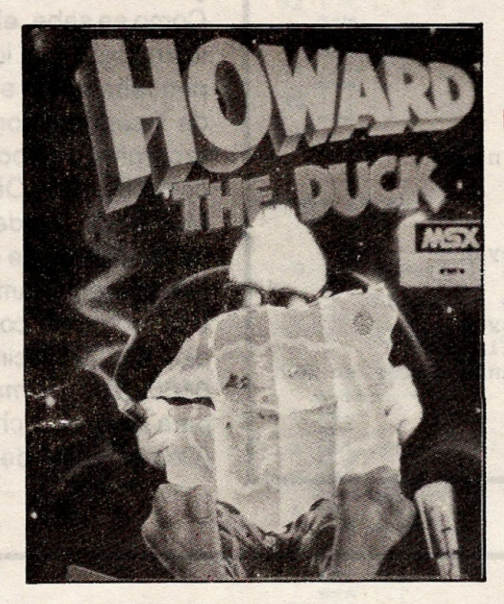
En la etapa inicial hay cuatro fases en donde el objetivo es aterrizar en el planeta y atravesar el pantano sobre el lomo de un ADREC CLONICO.

La primera fase es de aproximación al planeta. En la segunda se sobrevuela el área volcánica. En la tercera se penetra por un pasaje subterráneo que se conecta con el pantano y, finalmente, en la cuarta hay que aterrizar y capturar un ADREC con el que podremos atravesar el pantano.

En la segunda etapa hay que descender a lo más profundo de PHANTIS y atravesar seis niveles diferentes hasta rescatar al prisionero. En cada nivel nos aguardan diferentes peligros. Deberemos atravesar la superficie, la base alienígena, el bosque subterráneo y el lago interior para llegar al nivel MAGMA. Allí cruzaremos con un helicóptero la zona volcánica para llegar al nivel prisión. Cuando desintegremos a los diablos rojos que custodian la milenaria cárcel presenciaremos el final de PHANTIS.

Existen 24 enemigos diferentes en PHANTIS. Por cada uno que matemos obtendremos cierta cantidad de puntos. En la primera etapa contamos con 4 vidas y una extra cada 25000 puntos. En la segunda se comienza con 5 vidas y se obtiene una extra por cada corazón encontrado. (MICROBYTE)

HOWARD THE DUCK



Una aventura en la isla de Vulcano

Howard se halla en Vulcano. El reloj que se encuentra en la parte inferior de la pantalla muestra cuánto tiempo ha de transcurrir hasta que Vulcano entre en erupción. Tenemos que conseguir que Howard encuentre su equipo, derrote a varios mutantes, venza al Oscuro Overlord y rescate a Beverly y a Phill del interior de Vulcano.

Howard debe recoger su equipo saltando por sobre el pantano y atravesando la isla. Este equipo contiene un jet solar para cruzar canales, un ultraligero para llegar a la cumbre de la montaña volcánica y un desintegrador de neutrones portátil que le ayudará a enfrentarse contra el Oscuro Howard.

Con el jet solar cruzamos el canal y aparecerán los mutantes en la otra orilla.

A ellos los debemos derrotar con el quack-fu. Una patada a un mutante lo hiere y hace girar. Cuando está girando, se lo golpea de nuevo y el enemigo desaparece de la isla.

Cuando se atraviese el río y se elimine a los mutantes habrá que cruzar un puente colgante.

Luego Howard vivirá otros peligros y aventuras antes de llegar frente al oscuro Overlord. Hay que acabar con él y luego desactivar el volcán antes de que explote. (MICROBYTE)

POOYAN

Con arco y flechas debemos evitar que nos invadan los plowis. Estos son unos bichitos de aspecto simpático pero muy peligrosos.

Si dejamos que desciendan, se treparán por las escaleras y nos golpearán, quitándonos una vida.

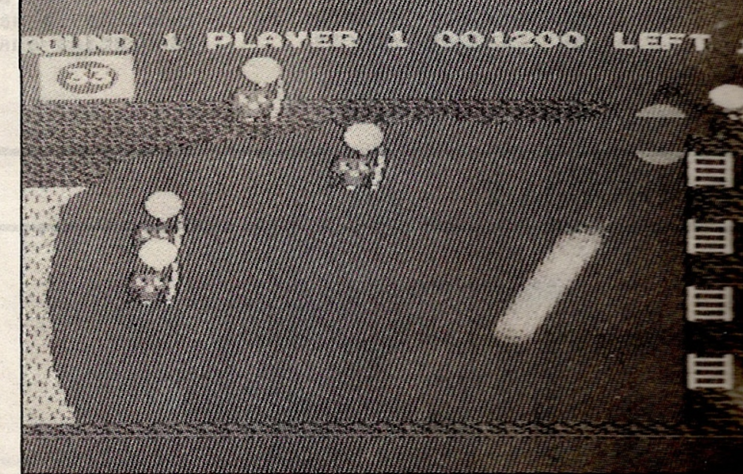
También nos disparan proyectiles mientras descienden (y tienen buena puntería).

Si nos llegan a dar con una de esas piedras también perderemos vidas.

La invasión se produce si dejamos que los globos lleguen al suelo. Por lo tanto debemos disparar con nuestro arco tratando de pinchar esos globos. Disponemos de dos clases de proyectiles: las flechas comunes y una cadena que aparece cada tanto y que nos permite eliminar a varios plowis juntos. También podemos ascender o descender a nuestro antojo con el ascensor.

Una vez finalizado el primer nivel se pasa a otro en donde los plowis tratan de alcanzar lo alto de la pantalla para empujar una piedra y hacerla caer sobre nuestra cabeza.

Así se siguen sucediendo los niveles. Cada dos tendremos una ronda de bonus en la que nos dedicaremos a obtener la mayor cantidad de puntos posibles. (REAL TIME)



TENTH FRAME

Este software es el mejor de los simuladores del juego de

Bowling hecho hasta el presente. Pueden jugar hasta ocho participantes, con dos opciones: por equipos o individualmente.

El juego se desarrolla con las reglas y elementos normales del bowling (bolas grandes). Nosotros debemos determinar la posición del jugador y la dirección del tiro.

Una vez lograda la ubicación deseada tendremos que realizar el tiro. Aquí hay que fijarse en la velocidad (SPEED), y el efecto o gancho (HOOK). Para un tiro perfecto debemos pulsar el joystick dos veces. Una, para determinar la velocidad, en cuanto el indicador llegue a la zona delimitada en pantalla con SPEED. Si pulsamos fuera de esa zona, puede salir un tiro no deseado. Para el efecto el mecanismo es similar, solamente que ahora hay que apretar el botón en la zona de HOOK.

Hay tres niveles de juego:

Novicios (KIDS): Cuando se tira la bola la velocidad se establece automáticamente y la bola irá recta si se manda sin gancho.

Amateur: Los tiros dependen de la fuerza con que se mande la bola y el efecto que se le dé.

Profesional: Aquí la exactitud en el uso del indicador velocidad/efecto es muy importante. Cualquier error en esto incrementa la probabilidad de un tiro malo.

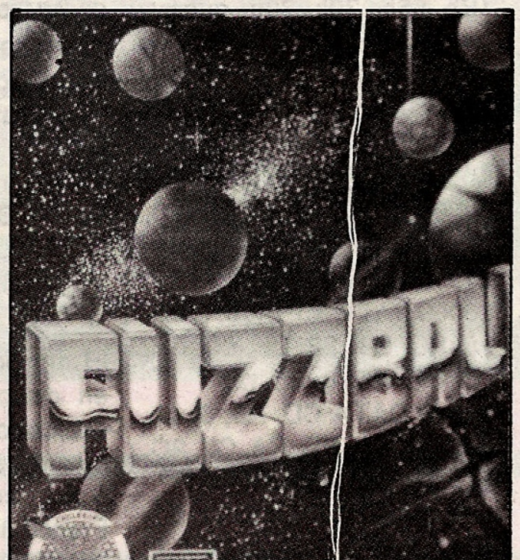
Por último nos resta señalar el procedimiento empleado por los autores del programa para lograr los movimientos de los jugadores. Se filmó el movimiento de un jugador profesional, y luego se traspasaron, cuadro por cuadro, los datos obtenidos a la computadora. Ese es el motivo de que nos parezca tan real este juego. (MICROBYTE)

FUZZBALL

Hay que saltar de un cubo a otro, a través de todos los cubos de la pirámide. La superficie de los cubos que ya hayamos visitado cambiará de color.

Una vez terminada la tarea en una pirámide se pasa a la siguiente. En los niveles superiores hay que visitar dos veces cada cubo para teñirlo del color correcto.

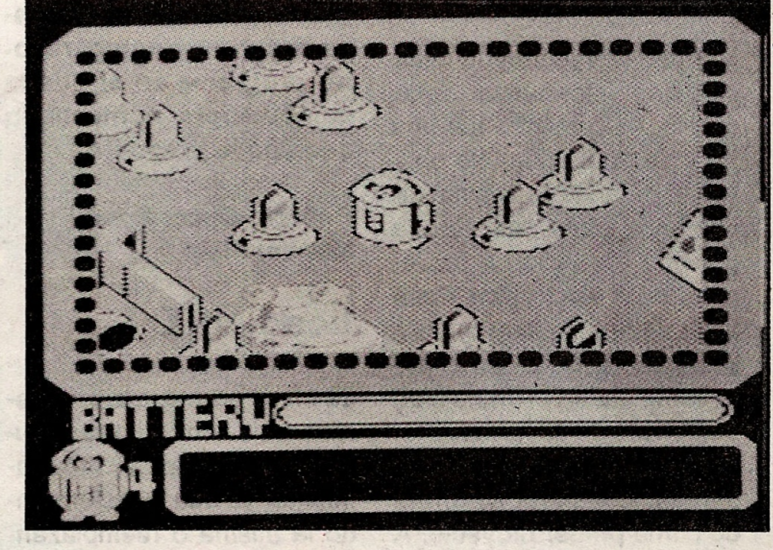
En cada pirámide hay cuatro ascensores que nos servirán para escapar de las bolas que caen de arriba o de las serpientes. A partir del nivel cinco comienzan a aparecer otros enemigos mucho más peligrosos.



De vez en cuando aparece una pelotita azul. Si logramos interceptarla congelaremos a nuestros enemigos durante algunos segundos. Ese tiempo lo debemos aprovechar para completar la pirámide.

Los puntos se obtienen de diversas formas. 10 puntos por cada salto, 50 por cada ascensor que no se use cuando la pirámide esté completa, 100 puntos por cada pelotita azul que se intercepte, 250 por interceptar al "mocoso de al lado" (un pequeño que se obstina en deshacer nuestro trabajo). Finalmente se obtiene un puntaje por completar cada pirámide: 1000 por la primera, 1500 por la segunda, 2000 por la tercera, etcétera.

Se logra una vida extra a los 10000 puntos y, a partir de allí cada 20000 puntos. (MICROBYTE)



MARTIANOIDS

Después de un largo viaje rutinario, MAKRON es atacado repentinamente por unos bárbaros alienígenas llamados MARTIANOIDS. Nuestra nave queda invadida por estos seres, afectando la transmisión de los programas al cerebro de la nave, y bajando peligrosamente el nivel de la batería al intentar los enemigos nuestra destrucción.

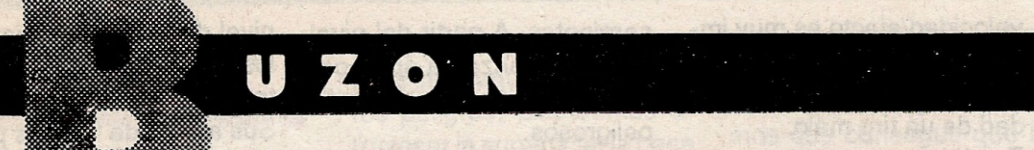
Sus armas de fotones pasan por las partes internas del cerebro, destruyendo casi todos los componentes activos.

Nuestra tarea es llevar el programa desde el transmisor hasta el receptor en cada sector. Cuando se recibe un programa se activan las defensas internas, impidiendo mayores daños en esa sección del cerebro. Hay que tener cuidado con los conductos de ventilación, ya que los programas se pueden caer por uno de esos conductos, y perderse para siempre.

Hay que reparar los componentes activos dañados usando los conos de repuesto. Hay pilas de repuesto por doquier, para usar cuando la carga baje a niveles peligrosos.

Para defendernos tenemos láseres que destruyen paredes internas, componentes activos, conos de repuesto y alienígenas. En la pantalla se nos muestra un mapa con nuestra posición, la del programa y la situación de cada sector.

El rojo indica un sector destruido y el blanco, uno activo. El rojo y amarillo parpadeante, un sector que está siendo atacado. También aparecen mensajes que nos indican qué está pasando en el cerebro. (REAL TIME)



MSX COMANDANTE

En el número de diciembre se publicó una nota que incluía una lista de materiales y circuito para la construcción de una interfase. Lamentablemente dicho plano era demasiado incompleto para que un principiante aficionado a la electrónica lo entendiera. De ser posible reimpriman el plano completo, con números de terminales. En el plano aparecía un terminal del slot designado como WA, ¿corresponde al WAIT o debió ser WR (WRITE)?

Horacio Morazzo
Azul

Load MSX

El esquema publicado tiene carácter informativo solamente, no pretende ser una nota de aplicación. No obstante, para los más avanzados en el tema, resulta una guía de aplicación e implementación electrónica. Pese a ser un circuito muy sencillo y elemental, requiere cierta experiencia en el manejo de periféricos de desarrollo propio y, por lo tanto, no es recomendable encarar su armado como primer proyecto. A modo informativo diremos que los datos de cada uno de estos usuales circuitos integrados pueden solicitarse en las casas más importantes del ramo electrónico que generalmente disponen de los manuales de datos correspondientes. Efectivamente, el terminal rotulado WA debe ser el WR (write), ya que se encarga de determinar la dirección en que fluyen los datos, esto es, del computador a la interfase o viceversa.

Para comunicarse con nosotros deben escribirnos a "Load MSX", Paraná 720 ,5º Piso, (1017), Capital Federal.

ORDENAMIENTO LENTO

He copiado el programa de la revista Nro. 9 página 28, y cuando pido el listado alfabético, tarda como cinco minutos en dar el listado con 70 fichas de 6 items.
1)¿Cómo podría hacer para acelerar el listado?
2)¿Quisiera saber por qué llegan tan tarde las revistas a mi ciudad?
3)He copiado el sintetizador de voces de la K64; quiero saber cómo se puede hacer para agregarle otro programa en el que la computadora conteste lo que uno le pregunte.

Roberto Oscar Soria
Mar del Plata

Load MSX

La clave para acelerar el listado reside en optimizar la rutina que realiza el ordenamiento, ya sea transformando la misma o reemplazándola por otra más rápida sin descartar la alternativa del lenguaje de máquina.
El retraso en la llegada de las revistas ya ha sido hablado con nuestros distribuidores y esperamos que sea corregido a la brevedad.
Un programa que conteste preguntas efectuadas en forma oral a la computadora ocuparía muchísimo espacio de memoria si se pretende una respuesta más o menos coherente, ya que además de una base de conocimientos debe existir un hardware y/o software analizador sintáctico de nuestras palabras. En conclusión, conformémonos viendo las fantasías (no tan lejanas) por televisión y cine.

HARDCOPY

Hemos recibido varias cartas con referencia al artículo sobre la rutina de copia de pantalla publicado en febrero pasado.
Debemos recordar que en el mismo se especifica el tipo de impresora (SEIKOSHA GP 550 A).Para verificar si este programa es compatible con otros modelos o marcas se deben comparar los comandos necesarios para poner en modo gráfico la impresora. Si estos no coinciden, se pueden cambiar en el programa y así adaptarlo.
Para la impresora original los comandos son 1Bh, 39h, 1Bh, 56h, 30h, 30h que se encuentran en las líneas 9C5D, 9C62, 9C98, 9C9D, 9CA2 y 9CA7 en el listado desensamblado y en (1), (2), (3), (4), (5) y (6) en los datas

```
60008 DATA 3E, 0, 1, 30, 0, 2A, F8, F7, BD, C2, 4F, 9C, 21, 1, 0, 9, 22, DB
60009 DATA 9C, 21, 0, 0, 3E, 18, BC, C8, 22, DF, 9C, 3E, (1), CD, A5, 0, 3E, (2)
60010 DATA CD, A5, 0, 11, 0, 0, 3E, F0, BB, CA, C7, 9C, ED, 53, DD, 9C, 2A, DF
60011 DATA 9C, 19, 11, E1, 9C, 1, 8, 0, CD, 59, 0, 11, 0, 0, 3E ,8, BB, CA
60012 DATA BA, 9C, 21, E1, 9C, 6, 8, 4E, CB, 11, 1F, 71, 2C, 10, F8, 47, 3E, (3)
60013 DATA CD, A5, 0, 3E ,(4) ,CD, A5 ,0, 3E, (5), CD, A5 ,0, 3E, (6), CD, A5, 0
60014 DATA 3A, DB, 9C, CD, A5, 0, 78, CD, A5, 0, 13, C3, 84, 9C, ED, 5B, DD, 9C
60015 DATA EB, 1, 8, 0, 9, EB, C3, 6A, 9C, 3E, D, CD, A5, 0, 3E, A, CD, A5
60016 DATA 0, 2A ,DF, 9C, 1, 0, 1 ,9 ,C 3, 56, 9C
```

Los dos primeros corresponden al espaciado entre líneas para modo gráfico y el resto es la predisposición para el modo gráfico; se deberán comparar con los que trae el manual de cada impresora y reemplazarlos si no coincidiesen.
De todos modos, para los que no se atrevan a modificar los bytes correspondientes, existe en el mercado un utilitario denominado GDUMP que realiza un hardcopy de pantalla en impresoras semiprofesionales.

CONSULTA MEDICA

Solicito de vuestra gentileza, me informen sobre el programa "HISTORIAS CLINICAS" de KAYAK. Me gustaría saber el lugar de adquisición, medio en que está grabado y precio.

Dr Alberto Porto
Rojas (Bs. As.)

Load MSX

El mencionado sistema viene en disco. El representante exclusivo en todo el país (hace envíos al interior) es MQA S.A., cuya dirección es French 3690 (Buenos Aires). Tel: 72-0856

ABRIL 1988

COMPUTACION K64 PARA TODOS

AÑO 4 Nº 37 ₳ 14 REP. ARGENTINA

DESDE EE.UU.
EL BOTIN DE LOS PIRATAS

COMMODORE
SUPER SOFT

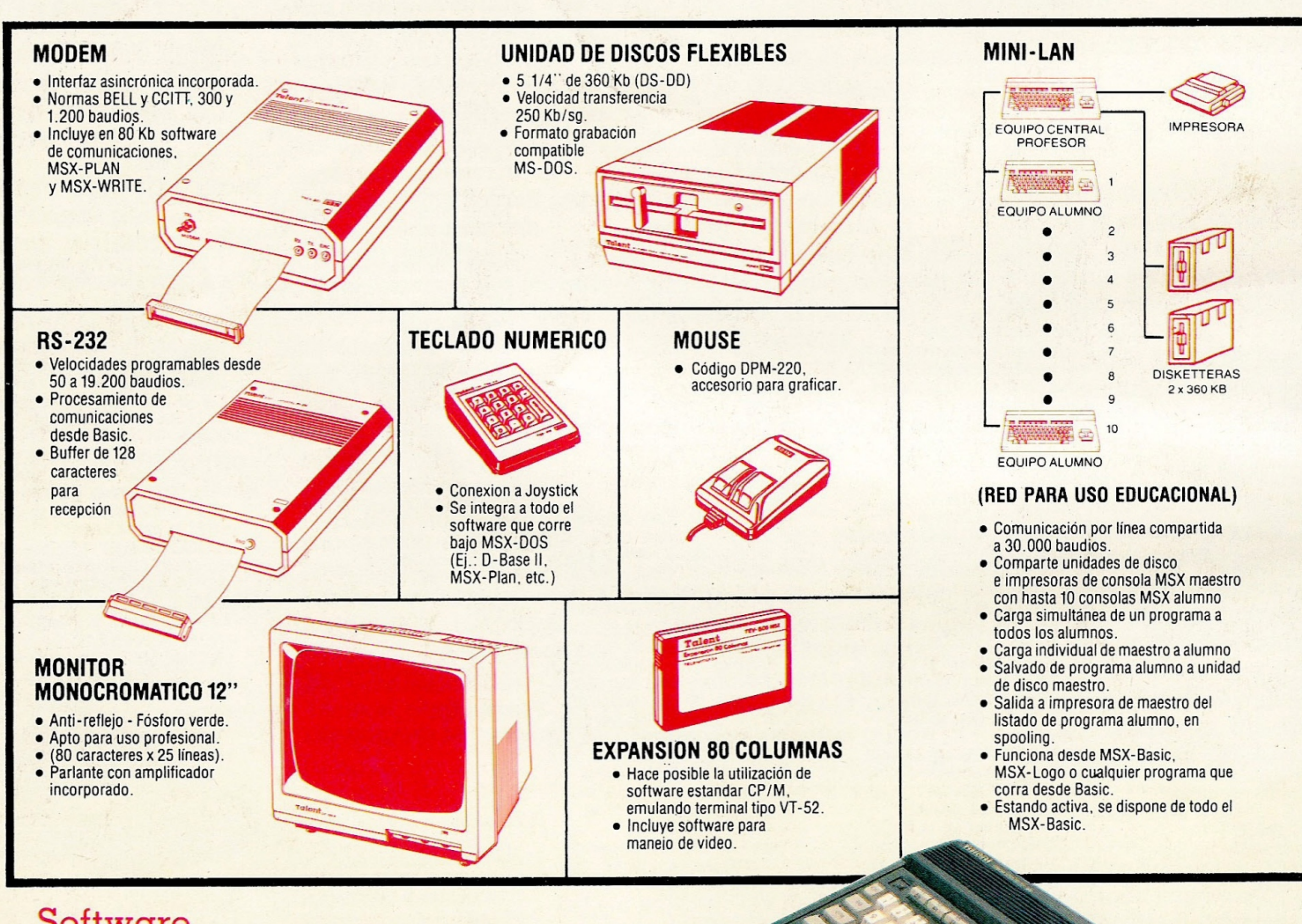
SPECTRUM
3 UTILITARIOS INEDITOS

B.B.S.
MODEM: COMO USARLO

MAS SORTEOS Y CONCURSOS

SPECTRUM - CZ - TK - DREAN COMMODORE - TI - MSX - ATARI y P

Encienda una computadora Talent MSX y sus periféricos.



Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.

Talent

Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.